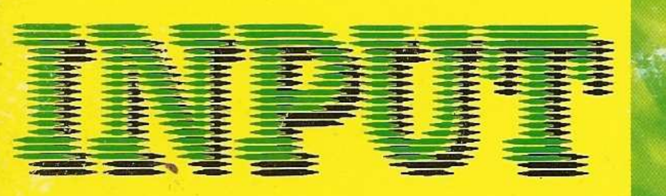


Publicación práctica para usuarios de

commodore

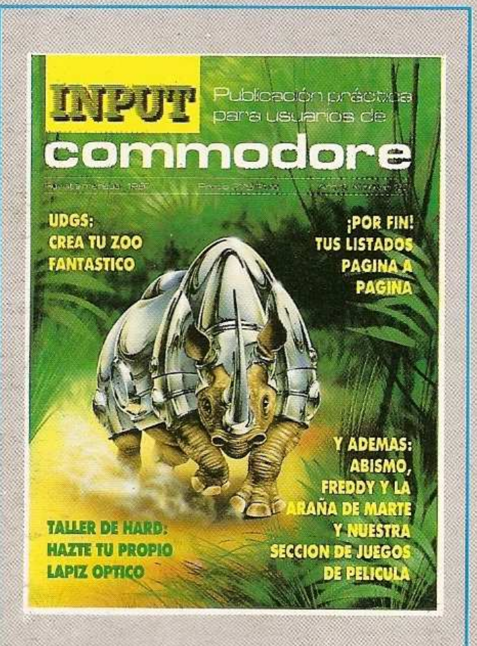
Revista mensual 1987 | Precio 375 Ptas | Año 2 Número 20

UDGS: CREA TU ZOO FANTASTICO

¡POR FIN! TUS LISTADOS PAGINA A PAGINA

Y ADEMAS: ABISMO, FREDDY Y LA ARAÑA DE MARTE Y NUESTRA SECCION DE JUEGOS DE PELICULA

TALLER DE HARD: HAZTE TU PROPIO LAPIZ OPTICO



AÑO 2 NUMERO 20

DIRECTOR: Manuel Pérez
DIRECTOR DE ARTE: Luis F. Balaguer
REALIZACION GRAFICA: Didac Tudela
COLABORADORES: Antonio Pliego, Xavier Ferrer, Josep M.ª Gils, Christophe Pais, Jaime Mardones, Equipo Molisoft, Carles Bartolomé, Ramón Ollé, Angels Alvarez
FOTOGRAFIA: Ernesto Walfisch, Joan Boada

INPUT Commodore es una publicación de PLANETA-DE AGOSTINI, S.A.
GERENTE DIVISION DE REVISTAS:
Sebastián Martínez

PUBLICIDAD: José Real-Grupo Jota
Madrid: c/ General Varela, 35
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: TECFA, S.A.

IMPRESION: Sirven Grafic
c/ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: B. 38.114-1986

SUSCRIPCIONES: EDISA
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Aribau, 185, 1.º
08021 Barcelona

DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Calle B, n.º 11. Sector B, Zona Franca
08004 Barcelona

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

INPUT Commodore es una publicación controlada por

INPUT Commodore es independiente y no está vinculada a Commodore Business Machines o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

INPUT commodore

SUMARIO

EDITORIAL

BUENOS MOMENTOS

Un cualificado representante de una de las principales empresas españolas distribuidoras de videojuegos nos comunicaba recientemente que el movimiento en esta franja del software para los microordenadores domésticos estaba superando todas las previsiones.

Evidentemente estas previsiones estaban referidas al reciente descenso de sus precios de venta al público.

Como ya habíamos comentado en estas mismas líneas en otras ocasiones, tal medida no debía limitarse a una simple operación de marketing destinada a invertir la relación entre el número de ejemplares vendidos y su precio unitario.

Así, el objetivo final debía ser el de definir una terapia de conjunto para el sector. Alcanzar un porcentaje de producción propia significativo, responder a las características específicas del mercado nativo, y fomentar el surgimiento de nutridos y sólidos grupos de programadores son un objetivo obligado de cualquiera que pretenda que el software en nuestro país deje de ser un vergonzoso, y exclusivo, problema de aduanas, licencias e impuestos.

En este cuadro, el papel que corresponde jugar a las revistas del sector debería modificarse también. Existe la necesidad de que los lectores de esas revistas y usuarios por tanto de los diferentes sistemas, identifiquen a los medios de comunicación como analistas veraces, críticos severos y cualificadores objetivos. Sólo cumpliendo estos requisitos serán portavoces reputados por sus valoraciones. Frente a los usuarios por su fiabilidad a la hora de inclinarse por una u otra opción. Frente a los productores y distribuidores por su análisis ajeno a la tendenciosidad o a una mal comprendida indulgencia.

Dicho esto, también debiera decirse que la problemática del software para nuestros sistemas no debe ni puede reducirse al subsector de los videojuegos. En un momento en que la informática invade más y más áreas de la actividad social y, entre ellas, la educativa, no debe pensarse sólo en fomentar una utilización unilateral de los pequeños, entre los más pequeños, micros.

Que productores y distribuidores se plantearan un apoyo consciente, técnico, económico y publicitario, a los diferentes tipos de software educativo y de aplicaciones sería también una manera de elevar el nivel de los usuarios y su capacidad de valorar críticamente sus propias actividades. Además de necesario sería muy sano.

LISTADOS PAGINADOS

Una de las funciones que normalmente se emplean al realizar un programa nuevo es el comando LIST, el cual nos sirve para ver las líneas de programa que ya hemos introducido e ir depurando y añadiendo nuevas líneas al mismo.

Uno de los grandes inconvenientes del LIST es la poca flexibilidad que posee cuando estamos buscando algún tipo de información dentro del programa y no sabemos exactamente dónde se encuentra.

Lo que normalmente se hace es listar desde el principio del programa, o bien a partir de una línea determinada sin especificar la línea final. A partir de aquí sólo es cuestión de ir pulsando la tecla de CTRL para ralentizar el desfile de líneas del programa por la pantalla, pero normalmente ocurre que nos damos cuenta de que el dato que buscábamos ya ha desaparecido por la parte superior de la pantalla y en este caso debemos pulsar la tecla de STOP y realizar otro nuevo LIST en las proximidades de donde lo hemos visto aparecer.

Una vez hemos conseguido localizar y detener el listado en la línea adecuada, observamos que necesitamos ver lo que se encuentra 20 o 30 líneas más arriba o más abajo, y de nuevo hay que hacer el correspondiente LIST con los problemas mencionados anteriormente.

Para solventar este inconveniente presentamos un programa en lenguaje ensamblador que nos permitirá avanzar y retroceder por todo el listado del programa con sólo pulsar una tecla. Este programa incorpora un nuevo comando al BASIC, y lo llamaremos LISTP porque realizará la función de presentarnos los listados por las páginas.

Esta ampliación está pensada para el Commodore 64 y para el C128 trabajando en modo Commodore 64.

SINTAXIS DEL NUEVO COMANDO

Existen cinco formas de utilizar el comando LISTP tal como puede verse en el siguiente ejemplo:

1. LISTP
2. LISTP 20
3. LISTP 20—50
4. LISTP —20
5. LISTP 50—

De estas cinco combinaciones sólo se usa la segunda para los listados paginados, aunque las restantes ofrecen las mismas prestaciones que un LIST normal sin dar error de sintaxis.

En la segunda forma, el número de línea indicado será el que encabeza la primera página del listado, siendo éste de longitud fija e igual a 10 líneas de programa.

Una vez ha aparecido la primera página es posible avanzar o retroceder números de línea con sólo pulsar las teclas '+' y '−'. Cabe mencionar que si en este avance llegamos a la última línea de programa, nos aparecerá una línea separadora, y el listado seguirá por la primera línea del programa, dando la sensación de lista circular. Esta particularidad también ocurre en el caso de que retrocedamos y lleguemos a la primera línea, en cuyo caso, antes de la línea separadora nos aparecerá la última línea.

Tanto el avance como el retroceso se realiza en incrementos de 5 líneas para permitir un solapamiento entre páginas consecutivas.

La forma de salir de este estado es pulsando la tecla de —STOP—.

En el caso de que el programa listado tenga menos de 5 líneas es posible que nos aparezca repetido varias veces el listado sin que ello presente problema alguno. En este caso es preferible hacer un LIST o un LISTP sin especificar número de línea.

El número de líneas listadas, así como los incrementos, es posible modificarlos a voluntad. Para ello deberéis consultar el apartado de modificaciones.

DESCRIPCION DEL PROGRAMA

El programa que presentamos aquí hace posible que el comando LIST del BASIC posea una gran mejora respecto al original. El nuevo comando se llamará LISTP, siendo el último carácter el que lo diferencia del LIST normal, de forma que nos recuerde fácilmente que se trata de presentar por páginas el listado del programa que estamos analizando.

En este caso no hemos usado la técnica del *wedge* para intervenir en la interpretación de los comandos del BASIC. En su lugar se han realizado las modificaciones sobre la misma rutina del LIST de forma que no sea necesario el tener que añadir algún carácter especial al inicio del nuevo comando. El programa que hace posible esto, una vez cargado en el ordenador, se ubicará en la zona de RAM situada entre la ROM del BASIC y el controlador del vídeo, de forma que nos quede aislado respecto a los programas que introduzcamos posteriormente. Como ya os habréis imaginado, esta zona empieza en la dirección 49152 ($C000) y es aquí donde empieza nuestra ampliación del comando LIST.

Para poder intervenir en la rutina del LIST original que se encuentra localizada en la ROM del BASIC es necesario obtener en primer lugar una réplica exacta del contenido de la ROM del BASIC en la RAM que se halla situada justo debajo de ella y que ocupa las mismas posiciones de memoria. Posteriormente es necesario introducir unos breves cambios del nuevo comando y actuar en conse-

cuencia en caso de que se trate del nuevo comando. Una vez se han introducido estas modificaciones, se actúa sobre la dirección número 1 para que la CPU pueda leer la RAM en lugar de la ROM.

Para poder analizar la posible simultaneidad con otros programas cabe destacar que la zona ocupada por nuestra ampliación va desde la dirección 49152 hasta la 49599 ($C000—$C1BF), ambas inclusive, y hace uso de las direcciones $FB y $FC de la página cero para las rutinas de copia.

DESCRIPCION DE LAS RUTINAS

En este apartado estudiaremos cómo funcionan las rutinas creadas y cuál es su efecto sobre la rutina original del LIST. Este estudio puede resultar un poco difícil a los lectores no iniciados en la materia, en tal caso puede pasarse al capítulo CARGADOR DE BASIC, introducir el programa en el ordenador y empezar a disfrutar de las ventajas del nuevo comando.

INICIO

La activación del nuevo comando se realiza desde el BASIC haciendo SYS 49152.

Esto hace que sea ejecutada la rutina de INICIO situada en la dirección 49152 ($C000), la cual ejecuta las subrutinas COPIAR, MODIF y ROMRAM antes de devolver el control al BASIC.

COPIAR

La subrutina COPIAR se encuentra situada a partir de la dirección $C014.

Consta de dos partes: la primera es la encargada de copiar la ROM del BASIC, y la segunda realiza lo mismo con la del KERNAL. Para realizar la copia lo que se hace es leer un byte y volver a escribir sobre él. Para ello es necesario preparar un puntero que nos indique en cada momento la dirección de copiar. Este puntero se encuentra localizado en las direcciones $FB y $FC y es inicializado con la dirección más baja de cada ROM a copiar, teniendo en cuenta que en la dirección $FB se encuentra el byte de menor peso y en $FC el byte de mayor peso. Para la ROM del BASIC este puntero es el $A000 y para la ROM del KERNAL es la $E000. Para realizar dicha copia usamos dos subrutinas casi iguales, de las cuales sólo vamos a comentar la primera de ellas.

Las instrucciones encargadas de hacer la copia (leer y escribir) son las que se encuentran en las direcciones $C01E y $C020, las cuales tienen un tipo de direccionamiento llamado indirecto-indexado por Y, siendo las que nos permitirán copiar bloques de 256 bytes modificando solamente el valor del registro Y (ver direcc. $C022). Debido a que la capacidad de cada una de estas dos ROM'S es de 8K bytes, será necesario copiar varios bloqueos de 256 bytes, y esto se consigue modificando el byte más alto del puntero, con la instrucción INC $FC situada en la dirección $C025 hasta llegar al final de cada ROM. Esto se consigue en las direcciones $C029 y $C03C, al comparar el puntero actual con la dirección siguiente a la del final de cada ROM. En el caso de la ROM del BASIC ésta será la $C000 y en el caso de la ROM del KERNAL será la dirección $0000.

MODIF

Esta subrutina es la encargada de modificar la rutina del LIST que ahora se encuentra en RAM e introducir, de esta forma, dos bifurcaciones o saltos hacia la nueva rutina del LISTP. Estas modificaciones se concretan en introducir un JMP $C083 en la dirección $A6A0 y un JMP $C062 en la dirección $A707. La primera dirección tiene por objeto el poder analizar los caracteres que entran después del comando original del LIST y decidir si debe ser un listado normal o paginado. La segunda dirección interviene en el número de líneas listadas. Si se trata de un listado paginado, la rutina del LIST sólo listará una línea y ésta estará bajo el control de la nueva rutina del LISTP.

ROMRAM

Esta subrutina se encuentra localizada a partir de la dirección $C076 y es la encargada de hacer la conmutación de ROM a RAM. Para que esta conmutación no sea perturbada por ningún evento extraño se realiza un bloqueo de las interrupciones mediante la instrucción SEI y al final son habilitadas de nuevo gracias a la instrucción CLI.

La conmutación se realiza modificando los bits 0 y 1 del port de la CPU cuya dirección es la 1.

La subrutina termina con un salto a la dirección $A474, que es la encargada de devolver el control al BASIC y de que nos aparezca el READY en pantalla.

MOD1 Y MOD2

La subrutina MOD1 es la encargada de listar sólo una línea si LISTP está activado y, en caso contrario, no afecta la operación normal del LIST.

Por su parte, la subrutina MOD2 está situada a partir de la dirección $C083 y actúa después de ser decodificado el comando LIST gracias a la bifurcación realizada en la subrutina MODIF.

Su misión es la de decodificar el nuevo comando con la sintaxis correcta y en caso de que éste no lo sea, dar el correspondiente mensaje de —sintax error—.

Tal como se ha descrito en el apartado de sintaxis del nuevo comando, éste sólo actúa en el caso que se especifique un sólo número de línea. Las tomas de decisión se realizan en la primera parte de esta subrutina (direc. $C083 a $C0A7).

LISTP

Esta subrutina empieza en la dirección $C0AA y es la encargada de realizar el listado de 10 líneas de programa y su funcionamiento es tal como se describe a continuación.

Empezamos con quitar la dirección de retorno del STACK, ya que la salida de nuestra subrutina debe devol-

ver el control al BASIC, y a continuación colocaremos el indicador (*flag*) LISPAG a 1 para que nos recuerde que entramos en modo paginado. Este *flag* se encuentra en la dirección $C00F y, cuando no está activa la rutina LISTP, su valor es cero.

A continuación guardamos la dirección donde se encuentra el puntero que nos indica donde empieza la línea siguiente a la solicitada en LINK1. Este puntero lo hemos llamado —*link*— porque es el encargado de ir encadenando las diferentes líneas de programa. Luego sigue un borrado de pantalla gracias a la utilización de la subrutina CHROUT ($FFD2) del KERNAL y la puesta a cero del contador de líneas listadas. A partir de este momento entramos en el bucle LISTP1 cuya misión es la de listar un total de 10 líneas de programa. Dentro de este bucle se realizan las siguientes comprobaciones: en primer lugar mira si ya ha listado la última línea de programa en cuyo caso debe trazar una línea separadora mediante la subrutina PRILIN y además debe colocar el puntero de la línea a listar sobre la primera línea de programa, es decir, al principio del área destinada para almacenar los programas en BASIC. En segundo lugar comprueba si hemos listado 5 líneas en cuyo caso guardaremos el nuevo puntero para su posterior utilización si escogemos la opción de avance.

El anterior bucle finaliza cuando se han listado 10 líneas de programa y a partir de este momento el programa espera una entrada desde teclado. La rutina EXPTEC es la encargada de realizar esta función y es capaz de analizar las teclas válidas para movernos por el listado del programa.

Tal como hemos mencionado anteriormente, podemos avanzar o retroceder con sólo pulsar las teclas '+' y '−' y salir del modo paginado con sólo pular la tecla de STOP.

La opción de avance tiene por objeto la de sustituir el puntero de la primera línea a listar por el de la quinta línea y regresar otra vez al bucle que nos lista las 10 líneas, LISTP1. De esta forma obtenemos un avance de 5 líneas permitiendo el solapamiento de otras 5 líneas.

La opción de retroceso tiene por objeto la de sustituir el puntero de la primera línea a listar por el de la quinta línea antes de la primera. Como que este puntero lo desconocemos, es necesario buscarlo, y ésta es la misión de la subrutina LKANT. A continuación regresamos al bucle de lisado LISTP1.

Por último, la opción de —STOP— borra la indicación (*flag*) de listados paginados y devuelve el control al BASIC.

LKANT

Esta subrutina empieza en la dirección $C157 y es la que se encarga de obtener la dirección donde se encuentra el puntero de la línea actual. Esta dirección se encuentra justo al inicio de la línea anterior a la listada.

Al igual que en el avance, tenemos que comprobar cuándo hemos llegado a un extremo del listado, en este caso es la primera línea. Si esto ocurre, lo que debemos hacer es desplazarnos hasta el final del programa y empezar a retroceder a partir de allí.

Esta operación se realiza 5 veces para poder retroceder otras tantas líneas y la operación termina con la actuación del puntero de la primera línea a listar.

Para los que deseéis profundizar en el tema se han incluido dos tablas que complementan el listado del programa en ensamblador, que son la tabla de símbolos y la de referencias cruzadas. En la primera se puede apreciar una lista de todas las etiquetas (*label*) usadas durante la confección del programa, junto con la dirección absoluta asignada a cada una de ellas. En la segunda tabla se observa una repetición de la tabla anterior ampliada con la información para saber cómo se accede a tal o cual etiqueta. Ambas tablas son de gran utilidad para este programa.

```
ØØØØ    ; * * * * * * * * * * * * * * * * * *
ØØØØ    ; *                                 *
ØØØØ    ; *       LISTADOS PAGINADOS        *
ØØØØ    ; *              por                *
ØØØØ    ; *      JOSEP M. GILS GIMENO       *
ØØØØ    ; *                                 *
ØØØØ    ; * * * * * * * * * * * * * * * * * *
ØØØØ    ;
ØØØØ    ; PARA ACTIVAR EL PROGRAMA DESDE ELBASIC DEBE
        HACERSE SYS 49152.
ØØØØ    ;
ØØØØ    ; LOS LISTADOS PAGINADOS SE OBTIENEN TECLEANDO
ØØØØ    ; —LISTP XXXX— SIENDO XXXX EL PRIMER NUMERO DE
        LINEA A VISUALIZAR
ØØØØ    ;
ØØØØ    ; CON LAS TECLAS '+' Y '−' PODREMOS LISTAR HACIA DELANTE O HACIA
        ATRAS
```

```
ØØØØ            ;
ØØØØ            ; CON INCREMENTOS DE 5 LINEAS. AL LLEGAR A LA
                ULTIMA LINEA APARECE UNA
ØØØØ            ;
ØØØØ            ; LINEA SEPARADORA Y EL LISTADO CONTINUA EN LA PRIMERA LINEA
ØØØØ            ;
ØØØØ            ; PARA SALIR DEBE PULSARSE LA TECLA DE —STOP—.
ØØØØ            ;
ØØØØ            ; EL PROGRAMA SE ALMACENA EN LA ZONA DE RAM
                SITUADA DESPUES DE LA ROM DEL
ØØØØ            ;
ØØØØ            ; BASIC (49152= $CØØØ).
ØØØØ            ;
ØØØØ            ; ****************************************************************
ØØØØ            ;
ØØØØ                       *=$CØØØ      ; ORIGEN DEL PROGRAMA.
CØØØ            ;
CØØØ  2Ø 14 CØ  INICIO JSR COPIAR       ; COPIA EL CONTENIDO DE LAS ROM'S SOBRE LA RAM.
CØØ3  2Ø 41 CØ         JSR MODIF        ; INSERTA MODIFICACIONES EN RUTINA DEL —LIST—.
CØØ6  4C 76 CØ         JMP ROMRAM       ; DEJAN DE TRABAJAR LAS ROM'S Y QUEDA ACTIVADO EL
                                        —LISTP—.
CØØ9            ;
CØØ9  Ø1 Ø8     LINK1  .WORD $Ø8Ø1
CØØB  ØØ ØØ     LINK5  .WORD $ØØØØ
CØØD  ØØ        LINLIS .BYTE $ØØ        ; NUMERO DE LINEAS LISTADAS POR PANTALLA (Ø-1Ø).
CØØE  ØØ        LATRAS .BYTE $ØØ        ; NUMERO DE LINEAS ANALIZADAS HACIA ATRAS (Ø-5).
CØØF  ØØ        LISPAG .BYTE $ØØ        ; FLAG DE LISTADOS PAGINADOS (ACTIVADO= $Ø1).
CØ1Ø  62 CØ     MODP1  .WORD
                MOD1
CØ12  83 CØ     MODP2  .WORD
CØ14            ;
CØ14            ; ****************************************************************
CØ14  A9 AØ     COPIAR  LDA #$AØ        ; COPIA LA ROM DEL BASIC SOBRE RAM.
CØ16  85 FC             STA $FC
CØ18  A9 ØØ             LDA #$ØØ
CØ1A  85 FB             STA $FB
CØ1C  AØ ØØ             LDY #$ØØ
CØ1E  B1 FB     LOOP1   LDA ($FB),Y
CØ2Ø  91 FB             STA ($FB),Y
CØ22  C8                INY
CØ23  DØ F9             BNE LOOP1
CØ25  E6 FC             INC $FC
CØ27  A5 FC             LDA $FC
CØ29  C9 CØ             CMP #$CØ
CØ2B  DØ F1             BNE LOOP1
CØ2D  A9 EØ             LDA #$EØ        ; COPIA LA ROM DEL KERNAL SOBRE RAM.
CØ2F  85 FC             STA $FC
CØ31  B1 FB     LOOP2   LDA ($FB),Y
CØ33  91 FB             STA ($FB),Y
CØ35  C8                INY
CØ36  DØ F9             BNE LOOP2
CØ38  E6 FC             INC $FC
```

```
CØ3A  A5 FC              LDA $FC
CØ3C  C9 ØØ              CMP #$ØØ
CØ3E  DØ F1              BNE LOOP2
CØ4Ø  6Ø                 RTS
CØ41             ;
CØ41             ; ******************************************************************
CØ41             ;
CØ41  A9 4C      MODIF   LDA #$4C    ; PREPARA LOS DESVIOS A LAS NUEVAS RUTINAS DEL
                                       —LISTP—.
CØ43  8D AØ A6           STA $A6AØ
CØ46  8D Ø7 A7           STA $A7Ø7
CØ49  AD 12 CØ           LDA MODP2
CØ4C  AE 13 CØ           LDX MODP2+1
CØ4F  8D A1 A6           STA $A6A1
CØ52  8E A2 A6           STX $A6A2
CØ55  AD 1Ø CØ           LDA MODP1
CØ58  AE 11 CØ           LDX MODP1+1
CØ5B  BD Ø8 A7           STA $A7Ø8
CØ5E  BE Ø9 A7           STX $A7Ø9
CØ61  6Ø                 RTS
CØ62             ;
CØ62             ; ******************************************************************
CØ62             ;
CØ62  AD ØF CØ   MOD1    LDA LISPAG  ; MIRA SI EL —LISTP— ESTA ACTIVADO.
CØ65  C9 Ø1              CMP #$Ø1
CØ67  FØ Ø7              BEQ MOD1A   ; SALTA SI ES ASI.
CØ69  AØ ØØ              LDY #$ØØ    ; SIGUE CON LA RUTINA ORIGINAL DEL —LIST—
CØ6B  B1 5F              LDA ($5F),Y
CØ6D  4C ØA A7           JMP $A7ØA
CØ7Ø  A9 ØD      MOD1A   LDA #$ØD    ; ENVIA UN RETURN.
CØ72  2Ø D2 FF           JSR $FFD2
CØ75  6Ø                 RTS
CØ76             ;
CØ76             ;  ****************************************************************
CØ76             ;
CØ76  78         ROMRAM   SEI        ; BLOQUEA INTERRUPCIONES IRQ.
CØ77  A5 Ø1               LDA $Ø1
CØ79  29 FD               AND #$FD
CØ7B  Ø9 Ø5               ORA #$Ø5
CØ7D  85 Ø1               STA $Ø1
CØ7F  58                  CLI        ; DESBLOQUEA INTERRUPCIONES IRQ.
CØ8Ø  4C 74 A4            JMP $A474
CØ83             ;
CØ83             ;  ****************************************************************
CØ83             ;
CØ83  C9 5Ø      MOD2      CMP #$5Ø  ; MIRA SI DESPUES DE —LIST— SIGUE LA LETRA 'P'.
CØ85  FØ Ø8                BEQ       ; SALTA SI ES ASI.
                           MOD2A
CØ87  C9 AB                CMP #$AB  ; MIRA SI DESPUES DE —LIST— SIGUE UN GUION.
CØ89  FØ Ø1                BEQ       ; SALTA SI ES ASI.
                           LINORM
```

```
CØ8B 6Ø       LERROR RTS          ; SALIDA POR ERROR DE SINTAXIS.
CØ8C 4C A4 A6 LINORM JMP $A6A4    ; SIGUE CON EL —LIST— NORMAL.
CØ8F 2Ø 73 ØØ MOD2A  JSR $ØØ73    ; LEE CARACTER SIGUIENTE A LA LETRA 'P'.
CØ92 9Ø Ø8           BCC MOD2B    ; SALTA SI ES NUMERICO.
CØ94 FØ F6           BEQ LINORM   ; SALTA SI NO HAY PARAMETROS
CØ96 C9 AB           CMP #$AB     ; MIRA SI ES UN GUION.
CØ98 DØ F1           BNE LERROR   ; SALTA SI NO ES ASI.
CØ9A FØ FØ           BEQ LINORM   ; SI LO ES REALIZA UN —LIST— NORMAL.
CØ9C 2Ø 6B A9 MOD2B  JSR $A96B    ; BUSCA PRIMER NUMERO DE LINEA.
CØ9F 2Ø 13 A6        JSR $A613    ; BUSCA PRIMERA DIRECC. DE LINK Y LEE SEGUNDO
                                    PARAMETRO.
CØA2 2Ø 79 ØØ        JSR $ØØ79    ; ANALIZA SEGUNDO PARAMETRO.
CØA5 FØ Ø3           BEQ LISTP    ; SALTA SI NO HAY MAS PARAMETROS.
CØA7 4C AF A6        JMP $A6AF    ; SALTA A LA RUTINA NORMAL DEL LIST SI HAY MAS
                                    PARAM.
CØAA                 ;
CØAA                 ; *******************************************************************
CØAA                 ;
CØAA 68       LISTP  PLA          ; QUITA LA DIRECC. DE RETORNO DEL STACK.
CØAB 68              PLA
CØAC A9 Ø1           LDA #$Ø1     ; COLOCA FLAG DE LISTADOS PAGINADOS A '1'.
CØAE 8D ØF CØ        STA LISPAG
CØB1 A6 5F           LDX $5F      ; GUARDA LINK PRIMERA LINEA.
CØB3 A5 6Ø           LDA $6Ø
CØB5 BE Ø9 CØ        STX LINK1
CØB8 8D ØA CØ        STA LINK1+1
CØBB A9 93    LISTP1 LDA #$93     ; BORRA PANTALLA.
CØBD 2Ø D2 FF        JSR $FFD2
CØCØ A9 ØØ           LDA #$ØØ     ; NUMERO LINEAS LISTADAS= Ø.
CØC2 BD ØD CØ        STA LINLIS
CØC5 AØ Ø1    LISTP2 LDY #$Ø1     ; COLOCA FLAG A 'NO STRING'
CØC7 84 ØF           STY $ØF
CØC9 B1 5F           LDA ($5F),Y  ; MIRA SI EL LINK ES DE LA ULTIMA LINEA +1.
CØCB DØ Ø8           BNE LISTP3   ; SALTA SI NO LO ES.
CØCD 2Ø 44 C1        JSR PRILIN   ; BUSCA LINK PRIMERA LINEA DE BASIC. TRAZA LINEA
                                    SEPARADORA.
CØDØ A9 ØD           LDA #$ØD     ; ENVIA UN RETURN
CØD2 2Ø D2 FF        JSR $FFD2
CØD5 C8       LISTP3 INY
CØD6 B1 5F           LDA ($5F),Y  ; BUSCA SIGUIENTE LINK
CØD8 AA              TAX
CØD9 C8              INY
CØDA B1 5F           LDA ($5F),Y
CØDC 2Ø E8 A6        JSR $A6EB    ; LISTA UNA SOLA LINEA.
CØDF EE ØD CØ        INC LINLIS   ; INCREMENTA EL NUM. DE LINEAS LISTADAS.
CØE2 A9 ØA           LDA #$ØA     ; MIRA SI SE HAN LISTADO 1Ø LINEAS.
CØE4 CD ØD CØ        CMP LINLIS
CØE7 FØ 17           BEQ EXPTEC   ; SALTA SI ES ASI.
CØE9 2Ø 32 C1        JSR LKSIG    ; BUSCA SIGUIENTE LINK.
CØEC A9 Ø5           LDA #$Ø5     ; MIRA SI SE HA LISTADO LA QUINTA LINEA.
```

```
CØEE  CD ØD CØ           CMP LINLIS
CØF1  DØ D2              BNE LISTP2    ; SALTA SI NO LO ES.
CØF3  A6 5F              LDX $5F       ; GUARDA EL LINK DE LA QUINTA LINEA.
CØF5  A5 6Ø              LDA $6Ø
CØF7  BE ØB CØ           STX LINK5
CØFA  BD ØC CØ           STA LINK5+1
CØFD  4C C5 CØ           JMP LISTP2    ; SIGUE CON OTRA LINEA.
C1ØØ            ;
C1ØØ  2Ø E4 FF  EXPTEC   JSR $FFE4     ; EXPLORA TECLADO.
C1Ø3  C9 2B              CMP #$2B      ; MIRA SI SE HA PULSADO LA TECLA '+'.
C1Ø5  DØ 13              BNE TEC1      ; SALTA SI NO LO ES.
C1Ø7  AE ØB CØ           LDX LINK5     ; COLOCA EL LINK5 EN EL LINK1.
C1ØA  AD ØC CØ           LDA
                LINK5+1
C1ØD  BE Ø9 CØ           STX LINK1
C11Ø  8D ØA CØ           STA
                LINK1+1
C113  86 5F              STX $5F
C115  85 6Ø              STA $6Ø
C117  4C BB CØ           JMP LISTP1    ; VUELVE A LISTAR 1Ø LINEAS.
C11A            ;
C11A  C9 2D     TEC1     CMP #$2D      ; MIRA SI SE HA PULSADO LA TECLA '-'.
C11C  DØ Ø6              BNE TEC2      ; SALTA SI NO ES ASI.
C11E  2Ø 57 C1           JSR LKANT     ; BUSCA EL LINK DE 5 LINEAS ANTERIORES A LA PRIMERA.
C121  4C BB CØ           JMP LISTP1    ; VUELVE A LISTAR 1Ø LINEAS.
C124            ;
C124  A5 91     TEC2     LDA $91       ; MIRA SI SE HA PULSADO LA TECLA DE —STOP—.
C126  C9 7F              CMP #$7F
C128  DØ D6              BNE EXPTEC    ; SI NO ES ASI VUELVE A EXPLORAR EL TECLADO.
C12A  A9 ØØ              LDA #$ØØ      ; BORRA FLAG DE LISTADOS PAGINADOS.
C12C  8D ØF CØ           STA LISPAG
C12F  4C 74 A4           JMP $A474     ; TERMINA EL —LISTP— Y REINICIALIZA EL BASIC.
C132            ;
C132            ;  *****************************************************************
C132            ;
C132  AØ ØØ     LKSIG    LDY #$ØØ      ; BUSCA LINK DE LA SIGUIENTE LINEA
C134  B1 5F              LDA ($5F),Y
C136  AA                 TAX
C137  C8                 INY
C138  B1 5F              LDA ($5F),Y
C13A  86 5F              STX $5F
C13C  85 6Ø              STA $6Ø
C13E  DØ Ø3              BNE LKSIG1    ; SI HA LLEGADO A LA ULTIMA LINEA,
C14Ø  4C 44 C1           JMP PRILIN    ; COLOCAR EL LINK A LA PRIMERA LINEA DEL BASIC.
C143  6Ø        LKSIG1   RTS
C144            ;
C144  A6 2B     PRILIN   LDX $2B       ; LINK PRIMERA LINEA DEL BASIC.
C146  A5 2C              LDA $2C
C148  86 5F              STX $5F
C14A  85 6Ø              STA $6Ø
C14C  A2 28              LDX #$28      ; VA A SACAR LA LINEA DE SEPARACION DE FINAL DE
```

```
                                        LISTADO.
C14E  A9 A8     PRILI1  LDA #$A8     ; CARACTER QUE FORMA LA LINEA DE SEPARACION.
C15Ø  2Ø D2 FF          JSR $FFD2
C153  CA                DEX
C154  DØ F8             BNE PRILI1
C156  6Ø                RTS
C157                    ;
C157                    ; ***************************************************************
C157                    ;
C157  A9 Ø5     LKANT   LDA #$Ø5     ; NUMERO DE LINKS HACIA ATRAS= 5.
C159  8D ØE CØ          STA LATRAS
C15C  AE Ø9 CØ  LKANØ   LDX LINK1    ; PREPARA PRIMER LINK.
C15F  AD ØA CØ          LDA LINK1+1
C162  86 5F             STX $5F
C164  85 6Ø             STA $6Ø
C166  E4 2B             CPX $2B
C168  DØ 18             BNE LKAN2    ; SALTA SI NO ES LA PRIMERA LINEA.
C16A  C5 2C             CMP $2C
C16C  DØ 14             BNE LKAN2    ; SALTA SI NO ES LA PRIMERA LINEA.
C16E  38                SEC          ; COLOCA EL LINK DE LA ULTIMA LINEA DEL BASIC.
C16F  A5 2D             LDA $2D
C171  E9 Ø2             SBC #$Ø2
C173  85 5F             STA $5F
C175  8D Ø9 CØ          STA LINK1
C178  A6 2E             LDX $2E
C17A  BØ Ø1             BCS LKAN1
C17C  CA                DEX
C17D  86 6Ø     LKAN1   STX $6Ø
C17F  8E ØA CØ          STX LINK1+1
C182                    ;
C182  C6 6Ø     LKAN2   DEC $6Ø      ; BUSCA EL LINK DE LA LINEA ANTERIOR.
C184  AØ FE             LDY #$FE
C186  B1 5F     LKAN3   LDA ($5F),Y
C188  C9 ØØ             CMP #$ØØ
C18A  FØ Ø3             BEQ LKAN5
C18C  88        LKAN4   DEY
C18D  DØ F7             BNE LKAN3
C18F  C8        LKAN5   INY
C19Ø  AD Ø9 CØ          LDA LINK1
C193  D1 5F             CMP ($5F),Y
C195  FØ Ø4             BEQ LKAN7
C197  88        LKAN6   DEY
C198  4C BC C1          JMP LKAN4
C19B  AD ØA CØ  LKAN7   LDA LINK1+1
C19E  C8                INY
C19F  D1 5F             CMP ($5F),Y
C1A1  FØ Ø4             BEQ LKAN8
C1A3  88                DEY
C1A4  4C 97 C1          JMP LKAN6
C1A7  88        LKAN8   DEY
C1A8  18                CLC
```

```
ØØ257  C1A9  98                   TYA
ØØ258  C1AA  65 5F                ADC $5F
ØØ259  C1AC  85 5F                STA $5F
ØØ26Ø  C1AE  8D Ø9 CØ             STA LINK1
ØØ261  C1B1  9Ø Ø2                BCC LKAN9
ØØ262  C1B3  E6 6Ø       ;        INC $6Ø
ØØ263  C1B5  A5 6Ø       LKAN9    LDA $6Ø
ØØ264  C1B7  8D ØA CØ             STA LINK1+1
ØØ265  C1BA  CE ØE CØ             DEC LATRAS
ØØ266  C1BD  DØ 9Ø                BNE LKANØ
ØØ267  C1BF  6Ø                   RTS
ØØ268  C1CØ                       .END

ERRORS= ØØØØ
```

```
SYMBOL VALUE
COPIAR  CØ14    EXPTEC  C1ØØ    INICIO  CØØØ    LATRAS  CØØE
LERROR  CØ8B    LINK1   CØØ9    LINK5   CØØB    LINLIS  CØØD
LINORM  CØ8C    LISPAG  CØØF    LISTP   CØAA    LISTP1  CØBB
LISTP2  CØC5    LISTP3  CØD5    LKANØ   C15C    LKAN1   C17D
LKAN2   C182    LKAN3   C186    LKAN4   C18C    LKAN5   C18F
LKAN6   C197    LKAN7   C19B    LKAN8   C1A7    LKAN9   C1B5
LKANT   C157    LKSIG   C132    LKSIG1  C143    LOOP1   CØ1E
LOOP2   CØ31    MOD1    CØ62    MOD1A   CØ7Ø    MOD2    CØ83
MOD2A   CØBF    MOD2B   CØ9C    MODIF   CØ41    MODP1   CØ1Ø
MODP2   CØ12    PRILI1  C14E    PRILIN  C144    ROMRAM  CØ76
TEC1    C11A    TEC2    C124

END OF ASSEMBLY
```

```
CROSS REFERENCE....PAGE 1


   COPIAR   $CØ14   3Ø   43
   EXPTEC   $C1ØØ  155  166  184
 * INICIO   $CØØØ   3Ø
   LATRAS   $CØØE   37  216  265
   LERROR   $CØ8B  111  117
   LINK1    $CØØ9   34  133  134  171  172  217  218  229  234  244  249  26Ø  264
   LINK5    $CØØB   35  162  163  169  17Ø
   LINLIS   $CØØD   36  138  152  154  158
   LINDRM   $CØ8C  11Ø  112  115  118
   LISPAG   $CØØF   38   85  13Ø  186
   LISTP    $CØAA  122  127
   LISTP1   $CØBB  135  175  18Ø
   LISTP2   $CØC5  139  159  164
   LISTP3   $CØD5  142  146
```

LKANØ	$C15C	217	266	
LKAN1	$C17D	231	233	
LKAN2	$C182	222	224	236
LKAN3	$C186	238	242	
LKAN4	$C18C	241	248	
LKAN5	$C18F	24Ø	243	
LKAN6	$C197	247	254	
LKAN7	$C19B	246	249	
LKAN8	$C1A7	252	255	
LKAN9	$C1B5	261	263	
LKANT	$C157	179	215	
LKSIG	$C132	156	191	
LKSIG1	$C143	198	2ØØ	
LOOP1	$CØ1E	48	51	55
LOOP2	$CØ31	58	61	65
MOD1	$CØ62	39	85	
MOD1A	$CØ7Ø	87	91	
MOD2	$CØ83	4Ø	1Ø7	
MOD2A	$CØ8F	1Ø8	113	
MOD2B	$CØ9C	114	119	
MODIF	$CØ41	31	7Ø	
MODP1	$CØ1Ø	39	77	78
MODP2	$CØ12	4Ø	73	74
PRILI1	$C14E	2Ø7	21Ø	
PRILIN	$C144	143	199	202
ROMRAM	$CØ76	32	97	
TEC1	$C11A	168	177	
TEC2	$C124	178	182	

CARGADOR DE BASIC

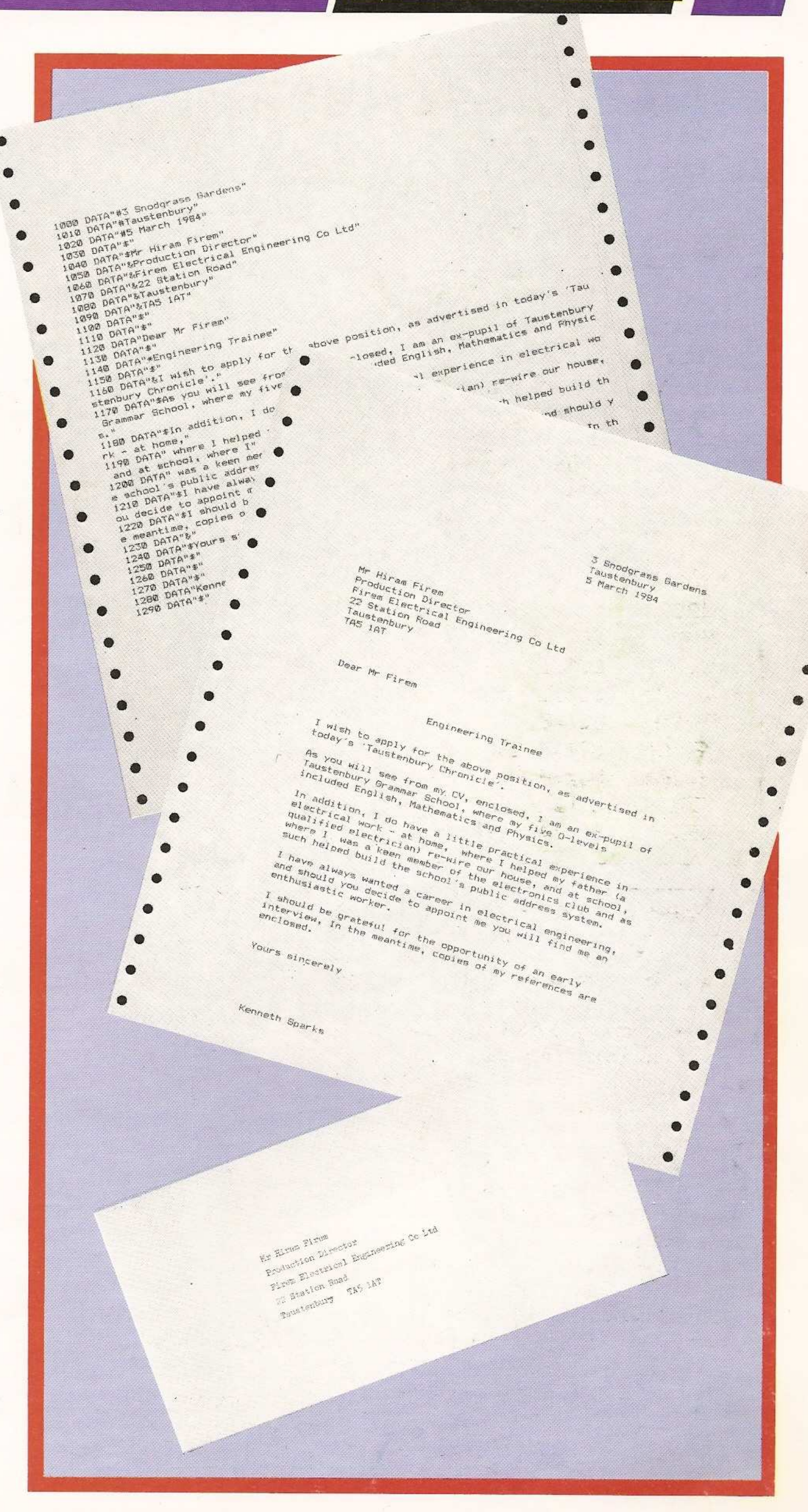
Para poder llevar a la práctica este programa de una forma asequible para vosotros, se ha pensado en usar un pequeño programa en BASIC que nos cargue el programa en lenguaje máquina a partir de la dirección 49152.

Este programa realiza la carga mediante la lectura de los valores introducidos en las sentencias DATA. Éstas se encuentran situadas en las líneas cuya numeración concuerda con la posición física que después tomará la rutina, es decir, de la 49152 hasta la 49599.

A continuación sale una presentación en pantalla de los comandos que se pueden utilizar y en el caso que se desee activar la rutina debe hacerse un SYS 49152. Para desactivar el comando LISTP basta con pulsar simultáneamente las teclas de —STOP— y de —RESTORE—.

```
10 REM * * * * * * * * * * *
20 REM *                   *
30 REM *     LISTADOS      *
40 REM *     PAGINADOS     *
50 REM *        por        *
60 REM *     JOSEP M.      *
70 REM *       GILS        *
80 REM *      GIMENO       *
90 REM *                   *
100 REM * * * * * * * * * * *
11Ø SUMA=Ø:AD=49152
12Ø FOR N=Ø TO 447
13Ø READ BY:POKE AD+N,
    BY:SUMA=SUMA+BY
14Ø NEXT
15Ø IF SUMA <>58358
    THEN PRINT "ERROR EN
    LOS DATOS":END
16Ø PRINT"[SHIFT+CLR/
    HOME][CTRL+9][11
    ESPACIOS]LISTADOS[2
    ESPACIOS]PAGINADOS[
    1ØESPACIOS][CTRL+Ø]"
17Ø PRINT"[4*CRSR
    ABAJO][CTRL+9]SYS
    49152[CTRL+Ø]ACTIVA-
    LISTPNUM.LINEA-"
18Ø PRINT"[CRSR
    ABAJO][CTRL+9]STOP/
    REST[CTRL+Ø]
    DESACTIVALISTP"
19Ø PRINT"[CRSRABAJO][8
    ESPACIOS][CTRL+9]+
    [CTRL+Ø]AVANZANUM.
    LINEA"
2ØØ PRINT"[CRSRABAJO][8
    ESPACIOS][CTRL+9]-
    [CTRL+Ø]RETROCEDE
    NUM.LINEA"
21Ø PRINT"[CRSRABAJO][5
    ESPACIOS][CTRL+9]
    STOP[CTRL+Ø]SALEDEL
    MODOPAGINADO"
49152 DATA32,2Ø,192,32,65,
      192,76,118
4916Ø DATA192,1,8,1,8,1Ø,Ø,Ø
49168 DATA98,192,131,192,
      169,16Ø,133,252
49176 DATA169,Ø,133,251,
      16Ø,Ø,177,251
49184 DATA145,251,2ØØ,2Ø8,
      249,23Ø,252,165
49192 DATA252,2Ø1,192,2Ø8,
      241,169,224,133
492ØØ DATA252,177,251,145,
      251,2ØØ,2Ø8,249
492Ø8 DATA23Ø,252,165,252,
      2Ø1,Ø,2Ø8,241
49216 DATA96,169,76,141,
      16Ø,166,141,7
49224 DATA167,173,18,192,
      174,19,192,141
49232 DATA161,166,142,162,
      166,173,16,192
4924Ø DATA174,17,192,141,8,
      167,142,9
49248 DATA167,96,173,15,
      192,2Ø1,1,24Ø
49256 DATA7,16Ø,Ø,177,95,
      76,1Ø,167
49264 DATA169,13,32,21Ø,
      255,96,12Ø,165
49272 DATA1,41,253,9,5,133,
      1,88
```

```
49280 DATA76,116,164,201,
        80,240,8,201
49288 DATA171,240,1,96,76,
        164,166,32
49296 DATA115,0,144,8,240,
        246,201,171
49304 DATA208,241,240,32,
        107,169,32
49312 DATA19,166,32,121,0,
        240,3,76
49320 DATA175,166,104,104,
        169,1,141,15
49328 DATA192,166,95,165,
        96,142,9,192
49336 DATA141,10,192,169,
        147,32,210,255
49344 DATA169,0,141,13,192,
        160,1,132
49352 DATA15,177,95,208,8,
        32,68,193
49360 DATA169,13,32,210,
        255,200,177,95
49368 DATA170,200,177,95,
        32,232,166,238
49376 DATA13,192,169,10,
        205,13,192,240
49384 DATA23,32,50,193,169,
        5,205,13
49392 DATA192,208,210,166,
        95,165,96,142
49400 DATA11,192,141,12,
        192,76,197,192
49408 DATA32,228,255,201,
        43,208,19,174
49416 DATA11,192,173,12,
        192,142,9,192
49424 DATA141,10,192,134,
        95,133,96,76
49432 DATA187,192,201,45,
        208,6,32,87
49440 DATA193,76,187,192,
        165,145,201,127
49448 DATA208,214,169,0,
        141,15,192,76
49456 DATA116,164,160,0,
        177,95,170,200
49464 DATA177,95,134,95,
        133,96,208,3
49472 DATA76,68,193,96,166,
        43,165,44
49480 DATA134,95,133,96,
        162,40,169,168
```



3 Snodgrass Gardens
Taustenbury
5 March 1984

Mr Hiram Firem
Production Director
Firem Electrical Engineering Co Ltd
22 Station Road
Taustenbury
TA5 1AT

Dear Mr Firem

Engineering Trainee

I wish to apply for the above position, as advertised in today's 'Taustenbury Chronicle'.

As you will see from my CV, enclosed, I am an ex-pupil of Taustenbury Grammar School, where my five O-levels included English, Mathematics and Physics.

In addition, I do have a little practical experience in electrical work - at home, where I helped my father (a qualified electrician) re-wire our house, and at school, where I was a keen member of the electronics club and as such helped build the school's public address system.

I have always wanted a career in electrical engineering, and should you decide to appoint me you will find me an enthusiastic worker.

I should be grateful for the opportunity of an early interview. In the meantime, copies of my references are enclosed.

Yours sincerely

Kenneth Sparks

Mr Hiram Firem
Production Director
Firem Electrical Engineering Co Ltd
22 Station Road
Taustenbury TA5 1AT

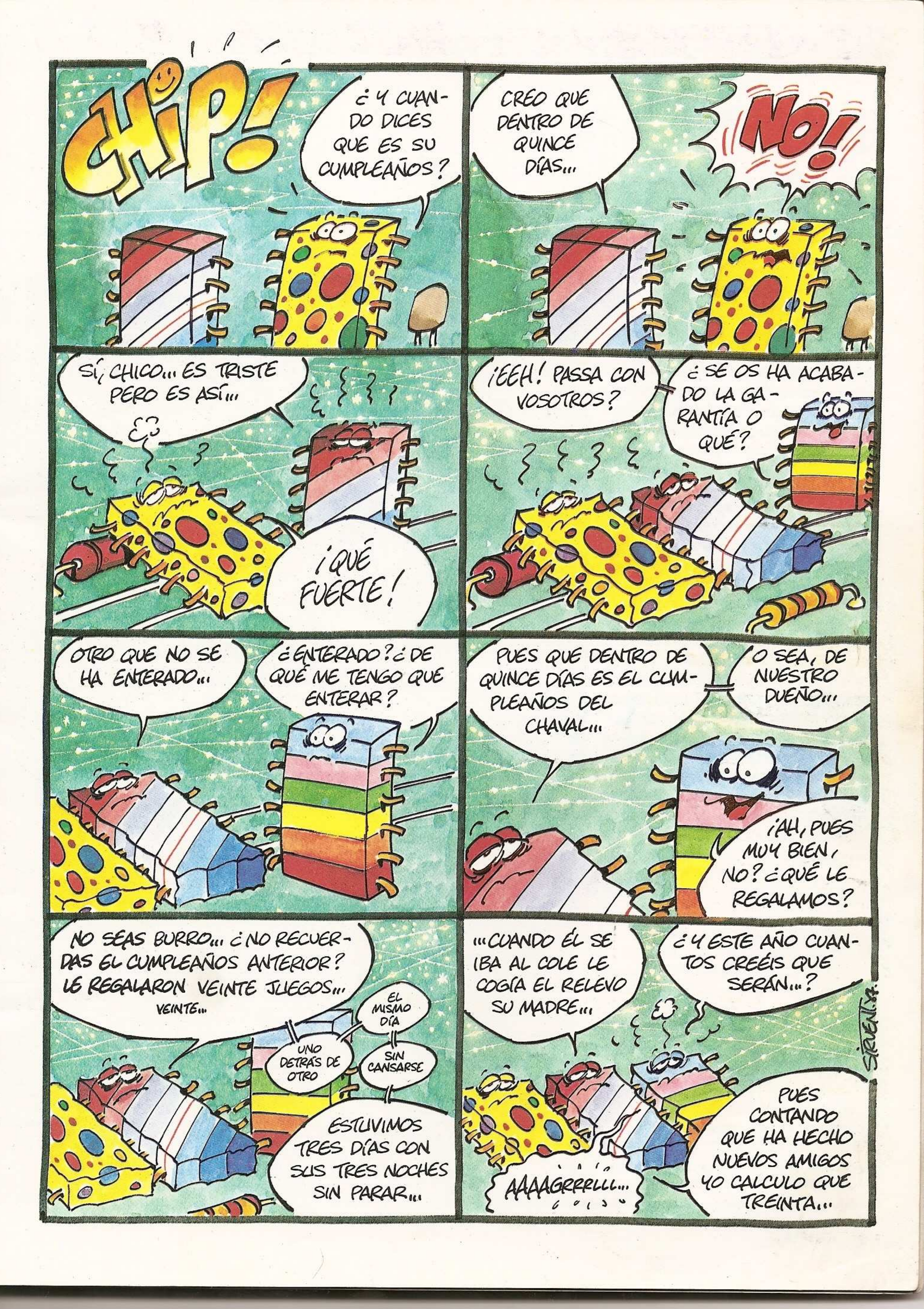
```
49488 DATA32,210,255,202,
        208,248,96,169
49496 DATA5,141,14,192,174,
        9,192,173
49504 DATA10,192,134,95,
        133,96,228,43
49512 DATA208,24,197,44,
        208,20,56,165
49520 DATA45,233,2,133,95,
        141,9,192
49528 DATA166,46,176,1,202,
        134,96,142
49536 DATA10,192,198,96,
        160,254,177,95
49544 DATA201,0,240,3,136,
        208,247,200
49552 DATA173,9,192,209,95,
        240,4,136
49560 DATA76,140,193,173,
        10,192,200,209
49568 DATA95,240,4,136,76,
        151,193,136
49576 DATA24,152,101,95,
        133,95,141,9
49584 DATA192,144,2,230,96,
        165,96,141
49592 DATA10,192,206,14,
        192,208,157,96
```

PARA INTRODUCIR MODIFICACIONES

Es posible modificar tanto el número de líneas listadas por página como los incrementos hacia delante o hacia atrás en el número de líneas. Si deseáis modificar estos valores es necesario consultar la tabla 1, donde se relacionan los tres conceptos mencionados.

TABLA 1

Concepto	Dirección	Valor
Núm. líneas listadas	49379	10
Increm. en avance	49389	5
Increm. en retroceso	49496	5

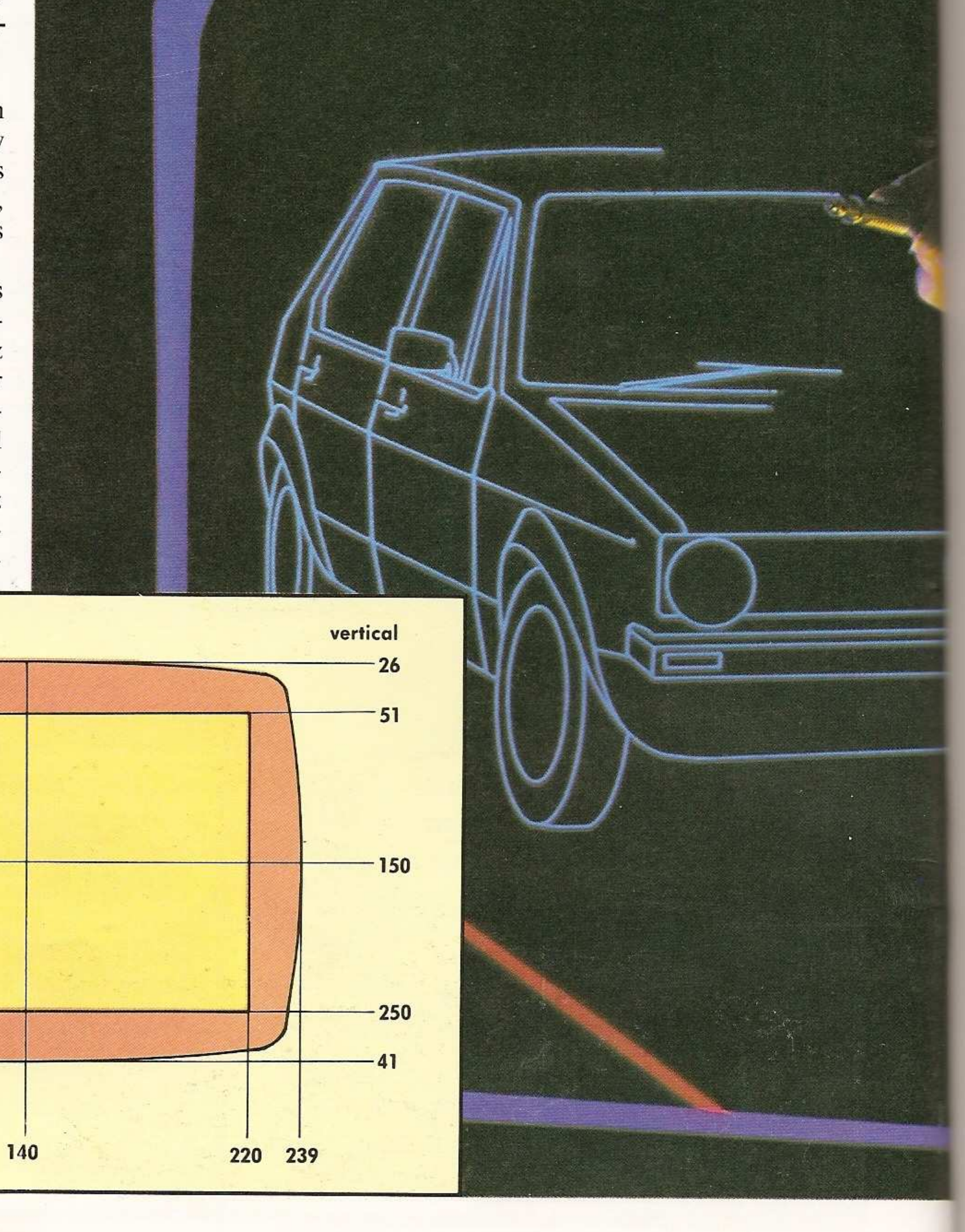

CHIP!
¿Y CUANDO DICES QUE ES SU CUMPLEAÑOS?
CREO QUE DENTRO DE QUINCE DÍAS...
NO!
SÍ, CHICO... ES TRISTE PERO ES ASÍ...
¡QUÉ FUERTE!
¡EEH! PASSA CON VOSOTROS?
¿SE OS HA ACABADO LA GARANTÍA O QUÉ?
OTRO QUE NO SE HA ENTERADO...
¿ENTERADO? ¿DE QUÉ ME TENGO QUE ENTERAR?
PUES QUE DENTRO DE QUINCE DÍAS ES EL CUMPLEAÑOS DEL CHAVAL...
O SEA, DE NUESTRO DUEÑO...
¡AH, PUES MUY BIEN, NO? ¿QUÉ LE REGALAMOS?
NO SEAS BURRO... ¿NO RECUERDAS EL CUMPLEAÑOS ANTERIOR? LE REGALARON VEINTE JUEGOS...
VEINTE...
EL MISMO DÍA
UNO DETRÁS DE OTRO
SIN CANSARSE
ESTUVIMOS TRES DÍAS CON SUS TRES NOCHES SIN PARAR...
...CUANDO ÉL SE IBA AL COLE LE COGÍA EL RELEVO SU MADRE...
¿Y ESTE AÑO CUANTOS CREÉIS QUE SERÁN...?
PUES CONTANDO QUE HA HECHO NUEVOS AMIGOS YO CALCULO QUE TREINTA...
AAAAGRRRLLL...
SIRVENT.

EL LAPIZ OPTICO

El lápiz óptico es uno de los periféricos más espectaculares de los que es posible dotar al Commodore 64. No obstante, como veremos en este artículo, su realización es muy sencilla, gracias a que nuestro ordenador ya tiene incorporados parte de los circuitos necesarios. También veremos unos útiles programas para sacar todo el jugo a este periférico.

El lápiz óptico es, en apariencia, un lápiz normal al que le falta la punta y del que sale un cable. Esta similitud es la que le da el nombre al dispositivo, aunque, como veremos, sus funciones son totalmente distintas.

Lo primero que hay que decir es que se trata de un dispositivo electrónico y óptico, ya que relaciona la luz que le llega de la pantalla del televisor con señales eléctricas que envía al ordenador. Estas señales informarán al ordenador del punto de la pantalla sobre el que se encuentra situado el lápiz en cada momento. Para poder proporcionar esta información el orde-





nador divide a nivel lógico la pantalla del televisor en una trama o retícula de pequeños puntos. Cada punto tiene asignadas dos coordenadas, una vertical y otra horizontal, de forma similar a como hace cuando trabaja en alta resolución. En la figura 1 se representan los valores que asigna a cada punto de la pantalla. El lápiz trabaja completamente con toda la pantalla del televisor, es decir, incluye tanto el fondo como el borde.

PRINCIPIOS DE SU FUNCIONAMIENTO

Como hemos dicho, el lápiz óptico se encarga de detectar la posición sobre la que se encuentra en la pantalla. Para ello se basa en el hecho de que las imágenes de televisión se forman al recorrer un haz de electrones la pantalla de línea en línea y de arriba hacia abajo. Al incidir el haz sobre un punto, éste se ilumina durante un instante pequeño de tiempo, aunque a nosotros nos parezca que permanece iluminado permanentemente debido a que nuestra retina es incapaz de seguir variaciones tan rápidas. El haz de electrones se mueve a una velocidad tal que pasa por toda la pantalla (formada por 625 líneas) cincuenta veces cada segundo. El lápiz, no obstante, sí es capaz de detectar el momento en que pasa el haz e ilumina el punto. En ese momento envía una señal determinada al ordenador, que es quien está controlando permanentemente la posición del haz a través de su chip de vídeo (el VIC 6569). De esta forma, sabiendo en qué punto está el haz en cada momento y conociendo en qué instante éste pasa por la posición donde se encuentra al lápiz, es posible determinar su ubicación en la retícula.

El *Commodore* posee una entrada específicamente destinada al lápiz que

se encuentra en el PORT 1 de juegos denominada LP (light pen); concretamente es el pin número 6, que coincide con la entrada del botón de disparo del joystick. Un nivel lógico «1», es decir, 5 V., indica al ordenador que el lápiz ho ha detectado nada; ahora bien, al pasar éste a «0» el ordenador interpreta que se ha encontrado el haz. El encargado de controlar esta señal es el Video Interface Controller, que es el circuito que maneja el haz. Al cambiar esta entrada de «1» a «0» el chip almacena los valores actuales del haz en dos registros internos, uno para la fila y otro para la columna. Estos registros conservan su valor hasta que se reciba una nueva transición y pueden ser leídos mientras tanto consultando las posiciones de memoria 53267 y 53268, correspondientes a los registros 19 y 20 del VIC.

Si se desea, al detectar el haz, el VIC puede generar una interrupción IRQ, con lo que pasará a ejecutar un programa en código máquina específico.

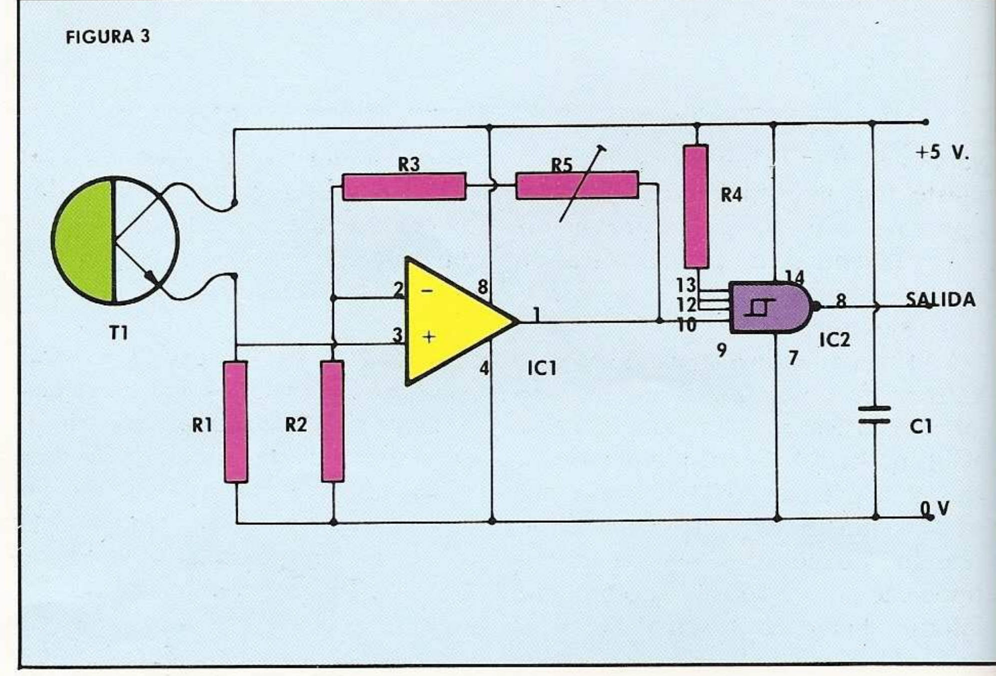
A través de la máscara de interrupciones del VIC se puede establecer si la interrupción se producirá o no. Posteriormente se ilustrará con un programa este modo de funcionamiento.

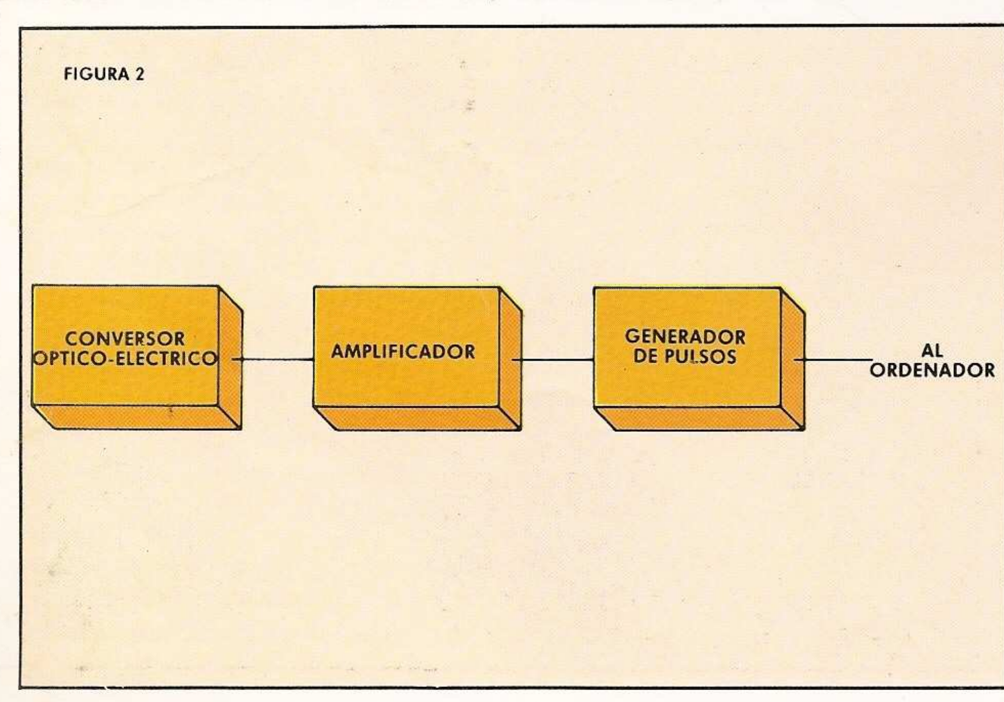
EL CIRCUITO ELECTRONICO

El componente principal de cualquier lápiz óptico es el transductor óptico-eléctrico, es decir, el elemento que es capaz de transformar variaciones luminosas en variaciones eléctricas de forma lo suficientemente rápida. Lo más adecuado por su bajo coste y buenas prestaciones es utilizar un fototransistor. El esquema teórico del circuito que se va a realizar se muestra en la figura 2.

El primer bloque es el encargado de convertir las señales luminosas en eléctricas y el elemento principal será el fototransitor. La señal que tendremos a la salida de este bloque será de un nivel bajo y, por consiguiente, será necesario amplificarla para adecuar el nivel a la siguiente etapa. Finalmente aparece un generador de pulsos cuya salida se conectará directamente al ordenador. Se hace necesario este tercer bloque para que los impulsos tengan una forma cuadrada bien definida e inteligible por el VIC.

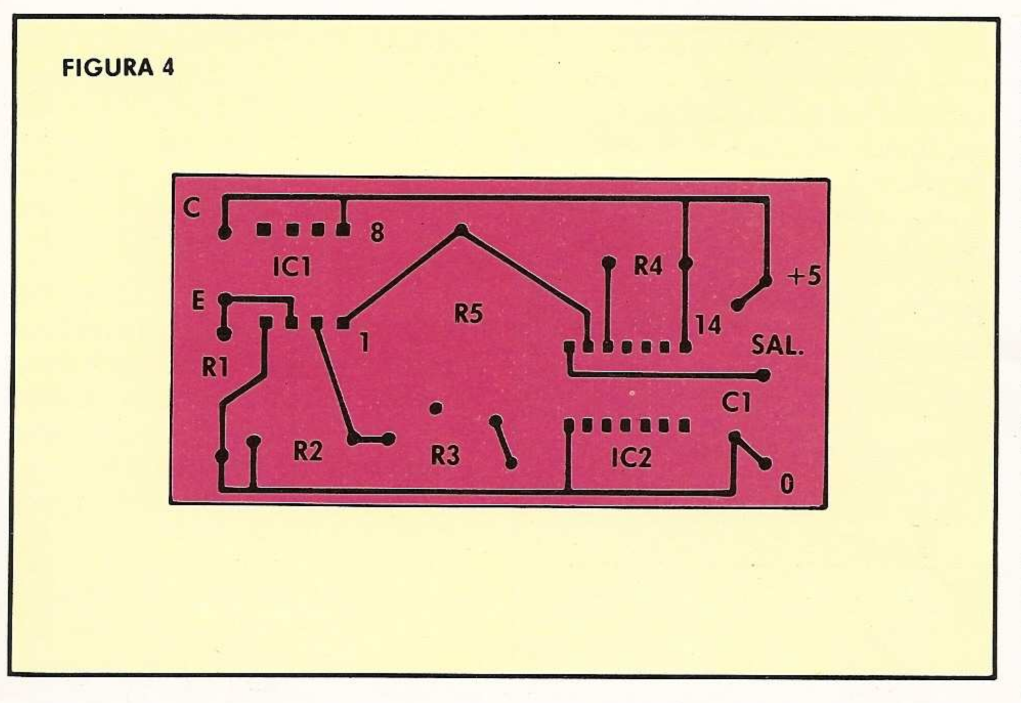
El circuito electrónico viene representado en la figura 3, en el cual se utilizan dos circuitos integrados que simplifican notablemente el montaje y lo hacen más fiable. Asimismo, en la figura 4 se muestra el circuito impreso a tamaño natural, visto por la cara de las pistas de cobre, sobre el cual se puede realizar el montaje de los distintos componentes.





El circuito, para aquellos que tengan inquietudes electrónicas, funciona del siguiente modo. El fototransistor actúa como un transistor normal NPN, con la diferencia de que la corriente que pasa por el colector no es controlada por la corriente de base, sino por la potencial luminosa que incide. Cuanta más luz reciba, mayor será la corriente que lo atraviese y por tanto mayor tensión habrá sobre la resistencia R1. Una tensión alta en R1 será indicativa de que se ha detectado el haz de la pantalla.

Esta tensión es amplificada por IC1, que no es más que un amplificador operacional. La ganancia de dicho amplificador está entre 4 y 6, pudiéndose



variar actuando sobre la resistencia de ajuste. Este margen de ganancias es el óptimo para un buen funcionamiento independiente de las características de luminosidad de la pantalla del televisor o monitor.

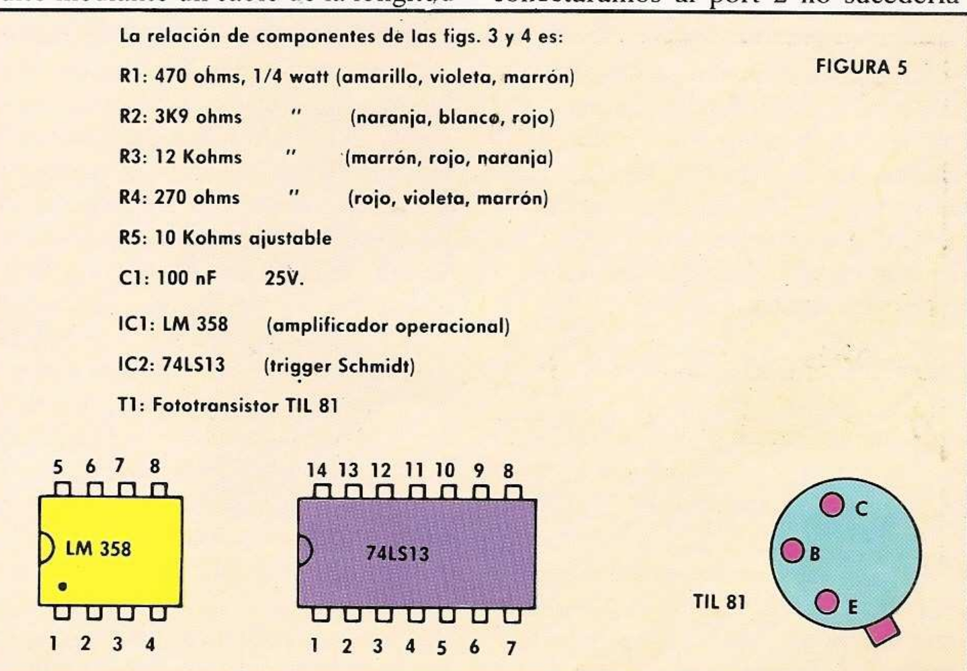
La señal amplificada actúa sobre la entrada de un trigger Schmidt. Este circuito básicamente proporciona a su salida un valor bajo cuando su entrada (formada por una AND de cuatro entradas) supera un cierto umbral de tensión. Cuando esta tensión de entrada cae por debajo de otro umbral, inferior al anterior, la salida pasa a un nivel alto. Cuando la señal de entrada vuelva a subir, la salida volverá a valer un nivel lógico «0» y así se repite el proceso. Obtenemos de esta forma a la salida un pulso cuadrado cada vez que el fototransistor detecte el haz, que es lo que nuestro ordenador espera del lápiz óptico.

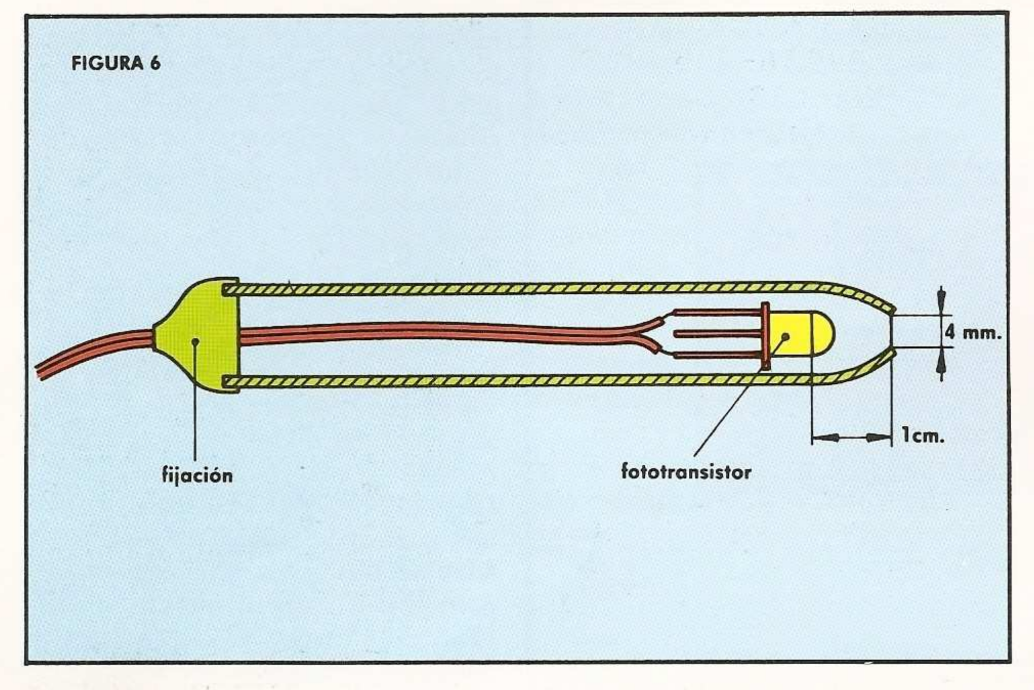
MONTAJE Y AJUSTE DEL LAPIZ

El montaje no presenta dificultades, y es adecuado incluso para aquellos lectores que no estén introducidos en el montaje electrónico. Para los que tengan dificultades a la hora de soldar o sean principiantes en estos avatares, es aconsejable que no suelden los circuitos integrados directamente sobre la placa sino que utilicen zócalos. Se debe prestar atención a la posición de los integrados y a las conexiones con el fototransistor y con el ordenador. Es conveniente empezar a soldar las resistencias y condensador primero, dejando los demás componentes para el final.

El fototransistor tiene tres terminales, pero sólo deben conectarse dos, el emisor y el colector, dejando libre el terminal de base. Éste no debe ir soldado a la placa de circuito impreso, ya que es el elemento que debe colocarse sobre la pantalla para detectar la luz. El fototransistor deberá, por tanto, conectarse con el resto del circuito mediante un cable de la longitud que deseemos (aproximadamente 1 metro). Para evitar que el fototransistor capte luz por los laterales que puede perturbar el buen funcionamiento del lápiz, deberá introducirse en un canuto, que puede ser el de un rotulador que previamente habremos vaciado. El diámetro interior debe ser lo suficientemente holgado como para que el fototransistor penetre sin dificultad, ya que se trata de un dispositivo sensible que no es conveniente forzar. La cara con la lupa debe situarse a una distancia de aproximadamente 1 cm de la punta del canuto. La punta de éste debe tener un diámetro de unos 4 mm y si es menor se hará necesario ensancharla para que penetre una cantidad suficiente de luz. Los cables de conexión irán a través del canuto y saldrán por el extremo contrario a la punta, consiguiéndose así una forma estética. Es conveniente fijar los cables de salida de alguna forma al canuto, para evitar que el fototransistor se mueva al manipular el lápiz, protegiéndolo además de posibles tirones ocasionales. Para esto es muy útil algún tipo de pegamento o adhesivo. En la figura 6 se representa esquemáticamente cómo debe construirse el lápiz.

El circuito debe conectarse al ordenador a través del PORT 1 de juegos, que es el único port que dispone de entrada para el lápiz óptico (si lo conectáramos al port 2 no sucedería





nada, pero no funcionaría). Del port debe extraerse la alimentación de 5 voltios y la toma de tierra, que se encuentran respectivamente en los pins 7 y 8. Además debe entregarse la información que dé el circuito a la entrada LP correspondiente con el pin 6. Por tanto, serán necesarios 3 hilos conductores entre el circuito montado y el ordenador. Es aconsejable que la longitud de dichos hilos sea corta, de menos de 10 cm, para así poder poner el circuito del lápiz cerca del ordenador sin tener demasiados hilos estorbando. En el comentario se encuentra sin ninguna dificultad el conector apropiado para el Port de juegos.

El buen funcionamiento del lápiz dependerá del brillo de la pantalla del televisor o monitor y del color de la misma. En principio el lápiz está diseñado para funcionar con un brillo normal que no moleste a la vista. Para probar el lápiz deberá ejecutarse el programa siguiente:

```
1Ø POKE 5328Ø, 1: POKE
   53281,1
2Ø PRINT PEEK (53267), PEEK
   (53268)
3Ø GOTO 2Ø
```

Lo que hace el programa es poner la pantalla en blanco, que es el color más luminoso, leer la posición de lápiz óptico e ir imprimiéndola. Para ajustar el lápiz lo situaremos en el centro de la pantalla y lo iremos desplazando lentamente hacia cualquier dirección. Si vemos que los valores imprimidos no varían deberemos actuar sobre la resistencia de ajuste girándola en sentido horario para así aumentar la ganancia del amplificador. Si aun así no obtenemos respuesta deberemos aumenter ligeramente el brillo del televisor hasta obtener variaciones en la lectura. Si incluso con un brillo fuerte de la pantalla derecha no se obtiene respuesta, se deberá desconectar el circuito y comprobar su montaje.

Con el programa de lectura adecuado que se incluye posteriormente, el lápiz tiene una resolución muy buena, aproximadamente de unos dos pixels en sentido horizontal, que es el más problemático. El peor comportamiento se observa en las cercanías de los bordes de la pantalla donde la curvatura de la misma es mayor y la luminosidad más pequeña. Para un correcto funcionamiento deberá ponerse cuidado en mantener el lápiz siempre perpendicular a la pantalla, ya que de otra forma cometería un mayor número de lecturas erróneas.

El color de la pantalla también influye en el comportamiento del lápiz debido a que el fototransistor no responde igual a todos los colores. Evidentemente el blanco, que es la mezcla de todos los colores, es el más adecuado. El lápiz también funciona bien con colores claros, sobre todo con la gama de azules, especialmente el cian, y de verdes. Es particularmente mala la gama de los rojos y naranjas. Usando colores distintos al blanco puede hacerse necesario aumentar un poco el brillo.

Por último sólo resta decir que el montaje del lápiz funcionará con el ordenador VIC-20 de la misma forma, sin tener que hacer ninguna modificación en el circuito. No sucederá lo mismo, como es lógico, con los programas, ya que están pensados para el Commodore 64. De todas formas no es demasiado difícil adaptarlos sabiendo que las posiciones de memoria donde se guardan las coordenadas del lápiz son las 36870 para la coordenada horizontal y la 36871 para la vertical.

RESOLUCION HORIZONTAL

Mientras que la resolución vertical del lápiz es perfecta, no ocurre así con la horizontal. Esto es debido a que el haz barre la pantalla horizontalmente y así, mientras en pasar de una línea a otra tarda 64 µs., es decir, 64 millonésimas de segundo, el tiempo que tarda en pasar de un punto a otro contiguo es mucho menor. Por este motivo, si leemos continuamente una posición fija del lápiz, veremos que oscila alrededor de un valor medio que es, precisamente, el correcto.

Habrá que solucionar este problema de alguna forma. Lo más adecuado será obtener un valor medio a partir de varias lecturas. Esto lo podemos hacer bien en BASIC o bien en código máquina, que será mucho más rápido en hacer la medida. Describimos a continuación un programa en C.M. que aprovecha la interrupción IRQ que se genera al detectar el haz, con lo que de esta forma es posible trabajar en BASIC y hacer la media simultáneamente de forma trasparente para el usuario. Explicaremos cómo puede hacerse esto para aquellos que se hayan sorprendido al leer que se pueden hacer varias cosas simultáneamente y para que así puedan utilizarlo en otras aplicaciones creadas por ellos mismos.

PROGRAMA PARA OBTENER EL VALOR MEDIO

En ensamblador:

```
CØØØ: SEI
CØØ1: LDA # $88
CØØ3: STA $DØ1A
CØØ6: LDA #$Ø8
CØØ8: STA $8D
CØØA: LDA #$16
CØØC: STA $Ø314
CØØF: LDA #$CØ
CØ11: STA $Ø315
CØ14: CLI
CØ15: RTS
CØ16: LDA $DØ13
CØ19: TAX
CØ1A: LDA $DØ19
CØ1D: STA $DØ19
CØ2Ø: AND #$Ø8
CØ22: BNE $CØ27
CØ24: JMP $EA31
CØ27: TXA
CØ28: CLC
CØ29: ADC $8B
CØ2B: STA $8B
CØ2D: LDA #$ØØ
CØ2F: ADC $8C
CØ31: STA $8C
CØ33: DEC $8D
CØ35: BNE $CØ4E
CØ37: LDY #$Ø3
CØ39: ROR $8C
CØ3B: ROR $8B
CØ3D: DEY
CØ3E: BNE $CØ39
CØ4Ø: LDA $8B
CØ42: STA $8E
CØ44: LDA #$ØØ
CØ46: STA $8B
CØ48: STA $8C
CØ4A: LDA #$Ø8
CØ4C: STA $8D
CØ4E: PLA
CØ4F: TAY
CØ5Ø: PLA
CØ51: TAX
CØ52: PLA
CØ53: RTI
```

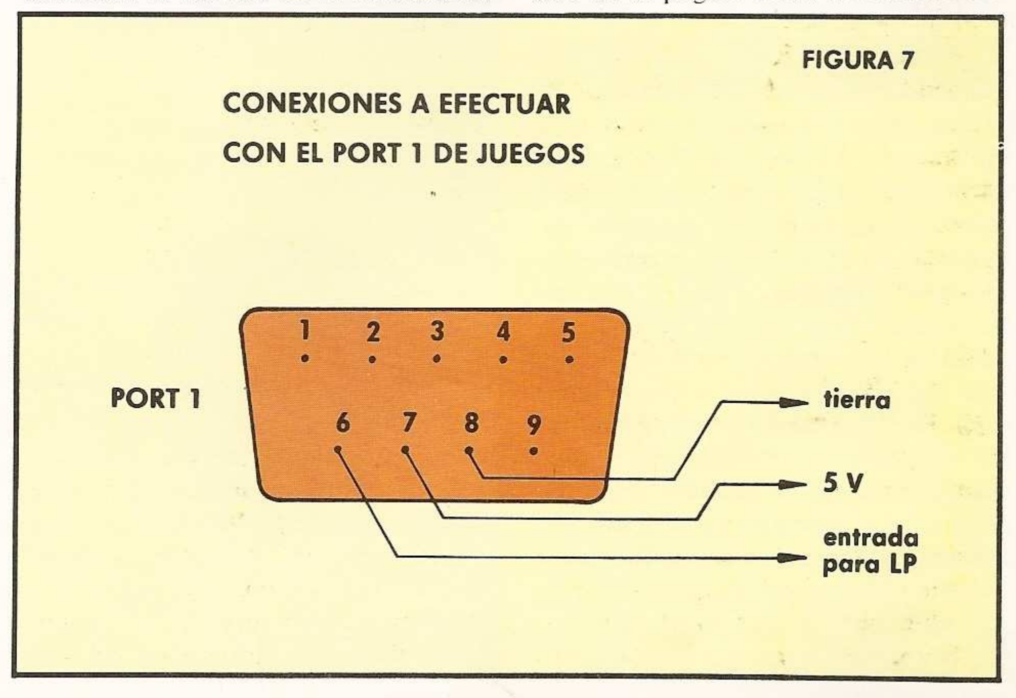
La versión BASIC es:

```
1Ø FOR I=49152 TO 49235 :
   READ A : POKE I,A : NEXT
2Ø DATA 12Ø,169,136,141,26,
   2Ø8,169,8,133,141,169,22,
   141,2Ø,3,169,192,141,21,3
3Ø DATA 88,96,173,19,2Ø8,
   17Ø,173,25,2Ø8,141,25,
   2Ø8,41,8,2Ø8,3,76,49,234,
   138
4Ø DATA 24,1Ø1,139,133,139,
   169,Ø,1Ø1,14Ø,133,14Ø,
   198,141,2Ø8,23,16Ø,3,1Ø2,
   14Ø
5Ø DATA 1Ø2,139,136,2Ø8,249,
   165,139,133,142,169,Ø,
   133,139,133,140,169,8,
   133
6Ø DATA 141,1Ø4,168,1Ø4,17Ø,
   1Ø4,64
```

El programa consta de dos partes bien diferenciadas. La primera parte se encarga de cambiar los vectores de salto de la interrupción IRQ (después se explicará qué significa esto), mientras que la segunda es la encargada de tomar las muestras y realizar la media, almacenando el resultado en una posición de memoria que luego podrá ser leída por cualquier programa.

La primera parte realiza las siguientes funciones. En primer lugar se inhiben las interrupciones para evitar que se pueda producir una interrupción mientras se están modificando los vectores de salto de la misma. A continuación se escribe en la máscara de interrupciones del VIC para indicarle que genere una interrupción cada vez que el haz sea detectado. Seguidamente pone un contador a 8, ya que realizaremos la media de 8 muestras, que será utilizado por la segunda parte del programa. Y, finalmente, se cambia el vector de salto de la IRQ. Este vector indica al ordenador dónde se debe saltar cada vez que reciba una IRQ. Normalmente este vector apunta a la posición $EA31, pero nosotros lo cambiamos para que salte a la segunda parte del programa. Hecho esto se permiten de nuevo las interrupciones y se retorna el control al BASIC. Ahora cada vez que se pida una IRQ el control pasará a nuestro programa.

La segunda parte es la que se encarga de hacer la media de las 8 muestras. En primer lugar carga en el registro X la coordenada horizontal del lápiz y en el acumulador carga el registro de interrupciones del VIC, comprobando si la IRQ ha sido generada por el lápiz o no, ya que puede ser generada también por otros circuitos. Si no ha sido pedida por el lápiz el control pasa a la rutina de IRQ normal, es decir, a la posición $EA31. En caso contrario se procede a ejecutar el programa que realiza las medias. El valor de la coordenada horizontal se almacena en dos posiciones consecutivas de la memoria, concretamente en $8By $8C de la página cero, acumulándolo



al valor que ya había en esas posiciones. Luego se decrementa el contador para ver si ya se han sumado los 8 valores; si es así hace la media dividiendo por 8 y la guarda en la posición $8E (142 en decimal), restaurando el contador a 8 y poniendo a 0 las posiciones $8B y $8C.

Para ejecutar el programa basta hacer SYS 49152. Aparentemente no pasa nada, pero nuestro programa ya está trabajando y mejorando considerablemente la resolución del lápiz en el eje X. El valor horizontal del lápiz se lee ahora en la posición 142 y no en la 53267.

Normalmente el lápiz generará 50 impulsos cada segundo, es decir, se producirá una IRQ y se tomará por tanto una muestra, cada 20 milésimas de segundo. Como necesitamos 8 muestras para obtener la media no podremos leer la posición horizontal del lápiz en intervalos de tiempo menores que 160 milésimas de segundo.

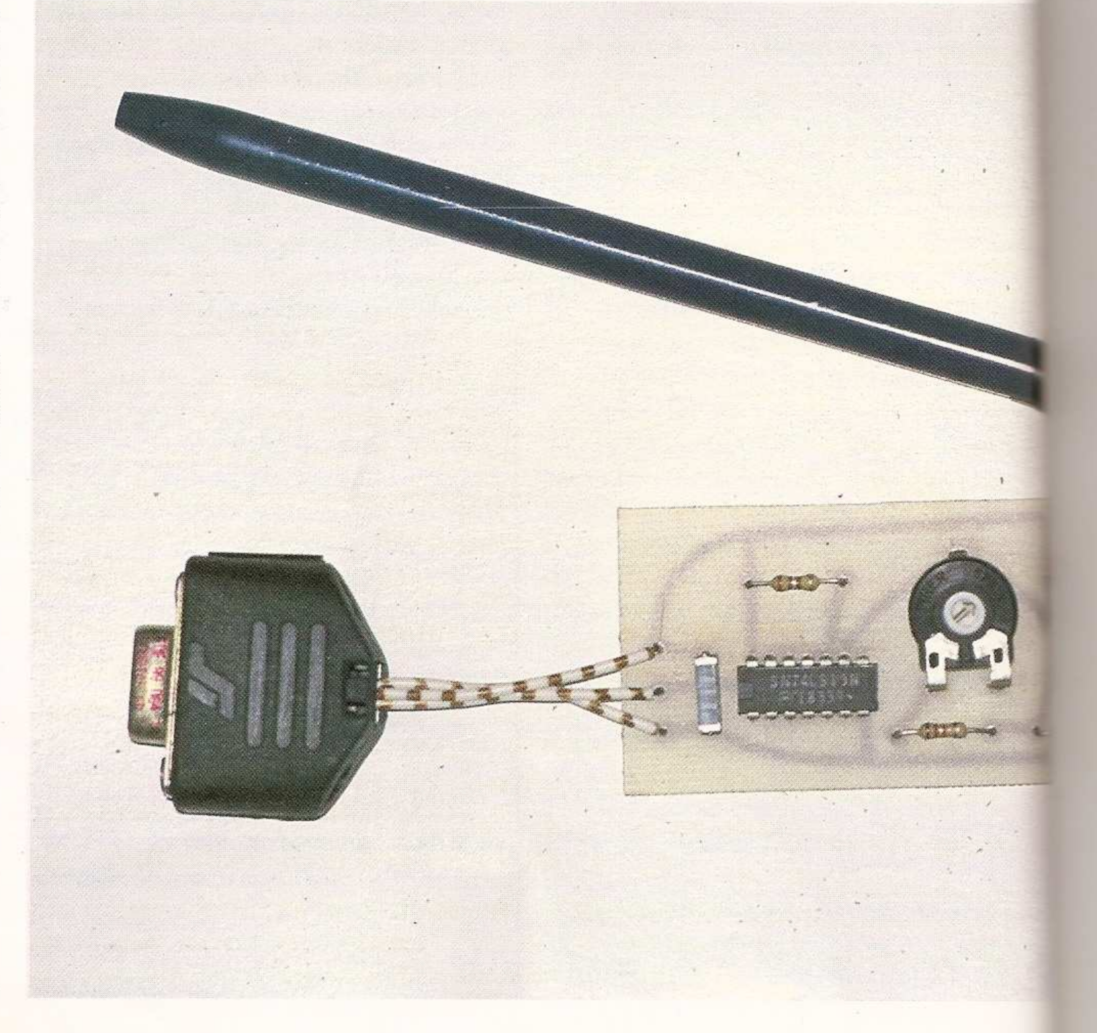
Hay que tener en cuenta que cada vez que se pulsan las teclas RUN/STOP y RESTORE, el vector de interrupciones se pone a su valor habitual, por lo que será necesario volver a hacer el SYS 49152 para reactivar nuestro programa.

En la coordenada vertical no es necesario realizar un proceso similar porque el valor obtenido directamente del registro del VIC ya es suficientemente exacto.

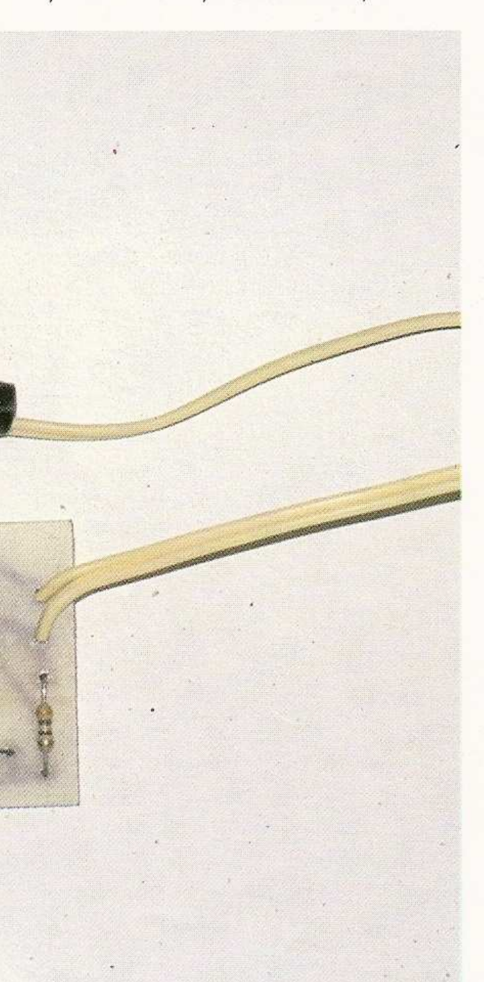
CALCULADORA EN PANTALLA

Una primera aplicación a la que el lápiz puede ir destinado es el control de programas y entrada de datos por pantalla. Así es posible ganar en ergonomía en el uso de programas. Un ejemplo ilustrativo, a la vez que espectacular, es el control de una calculadora que aparece en pantalla sin más que situar el lápiz sobre la tecla que se desea «pulsar». Permite controlar la entrada de datos numéricos mediante las «teclas» de números y el programa a través de las «teclas» de función que aparecen representadas.

```
10 SYS 49152: POKE 53281,
   11:POKE 53280,11:VA$ ="[
   14 ESPACIOS]"
15 REM****DIBUJO DE LA
   CALCULADORA****
20 PRINT"[SHIFT+CLR/HOME]
   "TAB(11)"[CTRL+1][CTRL+
   9][ 17 ESPACIOS][CTRL+
   0]"
30 FORI=1 TO 3:PRINT TAB(
   11)"[CTRL+9][CTRL+0][ 1
   5 ESPACIOS][CTRL+9] [CT
   RL+0]":NEXT
40 FOR I=1 TO 3:PRINT
   TAB(11)"[CTRL + 9][ 17
   ESPACIOS] [CTRL +
   0]":NEXT
50 PRINT TAB(11)"[CTRL + 9]
   [CTRL + 2]M+ [CTRL + 1]
   [CTRL + 2] M [CTRL + 1] [C
   T R L + 2]CE[CTRL + 1]
   [CTRL + 2]CLR[CTRL + 1] "
60 FOR I=1 TO 2: PRINT
   TAB(11)"[CTRL + 9] [ 17
   ESPACIOS][CTRL + 0]"
   :NEXT
70 PRINT TAB(11)"[CTRL + 9]
   [CTRL + 2]CLM[CTRL + 1]
   [CTRL + 2]SQR[CTRL + 1]
   [CTRL + 2]X12[CTRL + 1]
   [CTRL + 2] : [CTRL + 1] "
80 FOR I=1 TO 2: PRINT
   TAB(11)"[CTRL + 9] [ 17
   ESPACIOS][CTRL + 0]":NEXT
90 PRINT TAB(11)"[CTRL + 9]
   [CTRL + 2] 7 [CTRL + 1]
   [CTRL + 2] 8 [CTRL + 1]
   [CTRL + 2] 9 [CTRL + 1]
   [CTRL + 2] X [CTRL + 1] "
100 FOR I=1 TO 2:PRINT
    TAB(11)"[CTRL + 9][ 17
    ESPACIOS][CTRL +0]"
    :NEXT
110 PRINT TAB(11)"[CTRL+9]
    [CTRL+2] 4 [CTRL+1]
    [CTRL+2] 5 [CTRL + 1]
    [CTRL + 2] 6 [CTRL +1]
    [CTRL + 2] – [CTRL + 1] "
120 FOR I=1 TO 2:PRINT
    TAB(11)"[CTRL + 9][ 17
```



```
     ESPACIOS][CTRL +
     Ø]":NEXT
13Ø PRINT TAB(11)"[CTRL + 9]
     [CTRL + 2] 1 [CTRL + 1]
     [CTRL + 2] 2 [CTRL + 1]
     [CTRL + 2] 3 [CTRL + 1]
     [CTRL + 2] + [CTRL + 1] "
14Ø FOR I=1 TO 2:PRINT
     TAB(11)"[CTRL + 9][ 17
     ESPACIOS][CTRL +
     Ø]":NEXT
15Ø PRINT TAB(11)"[CTRL + 9]
     [CTRL + 2] Ø [CTRL + 1]
     [CTRL + 2]. [CTRL + 1]
     [CTRL + 2]+/-[CTRL + 1]
     [CTRL + 2] = [CTRL + 1] "
16Ø  PRINT TAB(11)"[CTRL +
     9][ 17 ESPACIOS][CTRL +
     0]"
17Ø WAIT653;1:X=PEEK(142):
     Y=PEEK(53268)
175 REM****IDENTIFICACION
     DE LA TECLA****
18Ø IF Y<12Ø AND Y>1Ø5
     THENTY=1
19Ø IF Y<139 AND Y>131
     THENTY=2
2ØØ IF Y<163 AND Y>155
     THENTY=3
21Ø IF Y<187 AND Y>179
     THENTY=4
22Ø IF Y<211 AND Y>2Ø3
     THENTY=5
23Ø IF Y<235 AND Y>227
     THENTY=6
24Ø IF X<122 AND X>1Ø6
     THENTX=1
25Ø IF X<135 AND X>123
     THENTX=2
26Ø IF X<151 AND X>139
     THENTX=3
27Ø IF X<17Ø AND X>154
     THENTX=4
28Ø IF TX=Ø OR TY=Ø
     THEN17Ø
29Ø FOR I=1 TO 3:POKE
     55463+12Ø*TY+4 *TX+I;
     2:NEXT
3ØØ GT=(TY-1)*4+TX
31Ø IF GT<13 THENON
     GTGOSUB 35Ø;36Ø;37Ø;
     38Ø;39Ø;4ØØ;41Ø;42Ø;
     44Ø;45Ø;46Ø;47Ø:GOTO
     33Ø
32Ø ON (GT-12)GOSUB 49Ø;
     5ØØ;51Ø;52Ø;54Ø;55Ø;
     56Ø;57Ø;59Ø;6ØØ;61Ø;62Ø
33Ø FOR I=1 TO 3:POKE
     55463+12Ø*TY+4 *TX+I;
     1:NEXT
34Ø TX=Ø:TY=Ø:IF FL=Ø
     THENGOSUB 72Ø
345 FL=Ø:GOTO 17Ø
347 REM****RUTINAS PARA
     LAS TECLAS****
35Ø ME=ME+NA:RETURN
36Ø NA=ME:RETURN
37Ø NA=Ø:ND=Ø:DP=Ø:RETURN
     38Ø  NA=Ø:AC=Ø: YA=Ø:
     DP=Ø: ND=Ø: RETURN
39Ø ME=Ø:RETURN
4ØØ NA=SQR(NA):RETURN
41Ø NA=NA12:RETURN
42Ø OP=4:YA=YA+1:IF YA=2
     THEN624
43Ø AC=NA:NA=Ø:AP=OP:FL
     =1:DP=Ø:ND=Ø:RETURN
44Ø NU=7:GOTO 7ØØ
45Ø NU=8:GOTO 7ØØ
46Ø NU=9:GOTO 7ØØ
47Ø OP=3:YA=YA+1:IF YA=2
     THEN624
48Ø AC=NA:NA=Ø: AP=OP:
     FL=1: DP=Ø: ND=Ø:
     RETURN
49Ø NU=4:GOTO 7ØØ
5ØØ NU=5:GOTO 7ØØ
51Ø NU=6:GOTO 7ØØ
52Ø OP=2:YA=YA+1:IF YA=2
     THEN624
53Ø AC=NA:NA=Ø: AP=OP:
     FL=1: DP=Ø: ND=Ø:
     RETURN
54Ø NU=1:GOTO 7ØØ
55Ø NU=2:GOTO 7ØØ
56Ø NU=3:GOTO 7ØØ
57Ø OP=1:YA=YA+1:IF YA=2
     THEN624
58Ø AC=NA:NA=Ø:
     AP=OP:FL=1: DP=Ø:
     ND=Ø: RETURN
59Ø NU=Ø:GOTO 7ØØ
6ØØ DP=1:RETURN
61Ø NA=-NA:RETURN
615 REM****TOTALIZACION DE
     OPERACIONES****
62Ø OP=AP:YA=1
624 IF AP<>OP
     THENIT=OP:OP=AP:AP=IT
63Ø ON OPGOTO 64Ø;65Ø;66Ø;
     67Ø
64Ø AC=AC+NA:GOTO 68Ø
65Ø AC=AC-NA:GOTO 68Ø
66Ø AC=AC*NA:GOTO 68Ø
67Ø AC=AC/NA
68Ø NA=AC:GOSUB
     72Ø:YA=YA-
     1:DP=Ø:ND=Ø:FL=1:IF
     GT<>24 THENNA=Ø
69Ø RETURN
7ØØ IF DP THENND = ND+1:
     NA=NA+NU/1Ø ↑ ND:RETURN
     71Ø  NA=NA*1Ø+NU:RETURN
     715  REM****RUTINA DE
     IMPRESION****
72Ø PRINT
     "[CTRL+2][CLR/HOME][
     2*CRSR ABAJO]" TAB(12);
```



```
    VA$:PRINT "[CLR/HOME][
    2*CRSR ABAJO]" TAB(12);
    LEFT$(VA$;15-
    LEN(STR$(NA)))+STR$(NA)
730 RETURN
```

El programa ejecuta todas las funciones que tiene una calculadora corriente:

—Suma, resta, multiplica y divide;
—permite el cambio de signo;
—eleva al cuadrado y extrae raíces;
—posee una memoria acumulativa;
—notación exponencial cuando el resultado excede de 12 dígitos.

Debido a la estructura modular del programa es posible modificar alguna de estas funciones e incluir otras, según las necesidades o gustos de cada uno.

Para manejarla basta con posicionar el lápiz sobre la tecla que aparece en pantalla y que se desea «oprimir». Ésta es normalmente de color blanco, ya que es el más adecuado para el lápiz. Cuando el lápiz está posicionado en la tecla basta con apretar la tecla SHIFT del lado izquierdo para indicar al ordenador que hemos hecho una selección. Inmediatamente éste responde poniendo la tecla de color rojo durante un instante, indicándonos así que la tecla ha sido aceptada. Debe procurarse situar el lápiz en la parte blanca de la tecla y no sobre la leyenda, ya que al ser de un color más oscuro podrían producirse errores.

Las teclas de la calculadora tienen asignada la siguiente función:

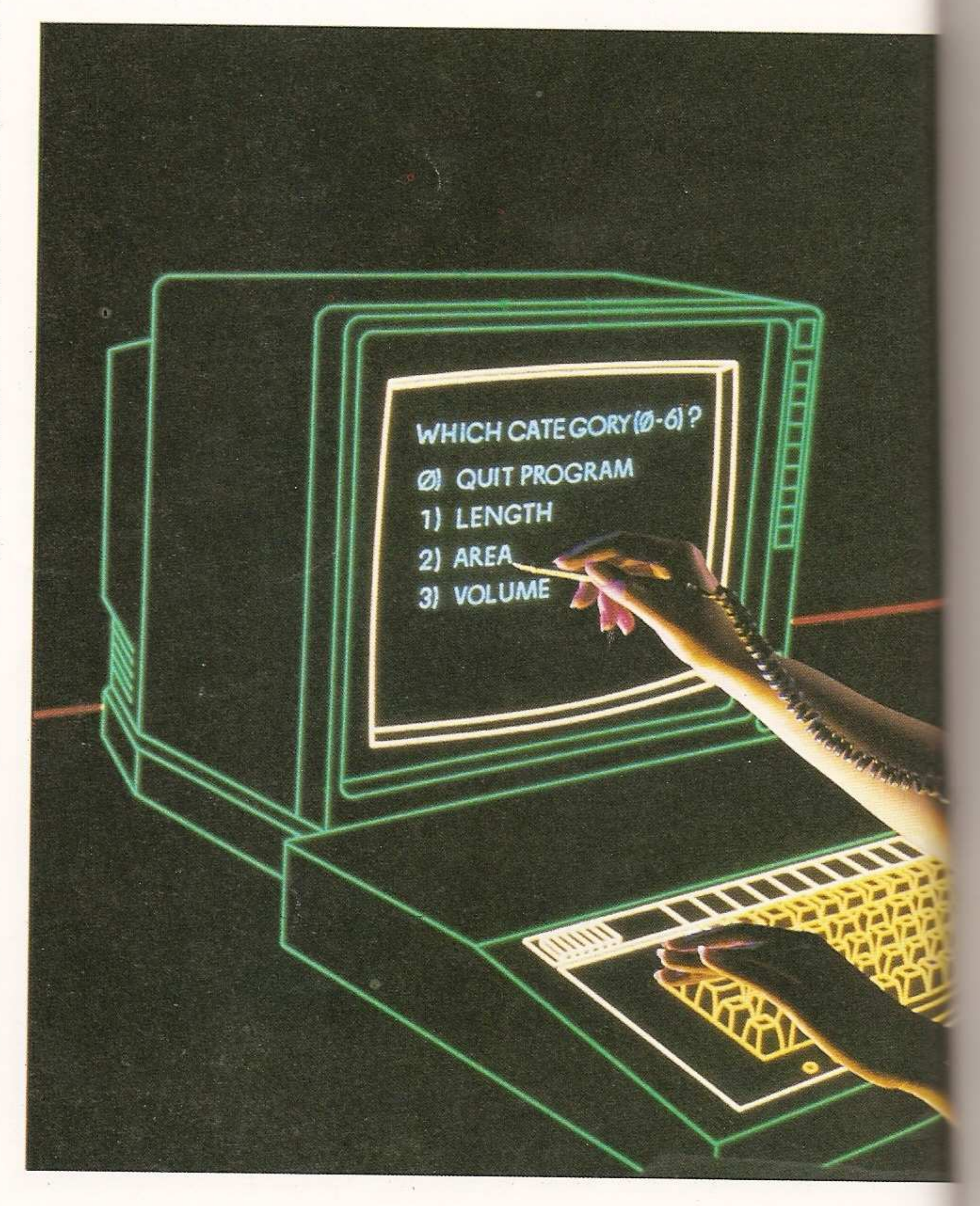
M +	Guarda en memoria el número que aparece en el display, sumándolo a lo que ya hubiera.
M	Muestra en el display el contenido de la memoria, borrando del mismo lo que hubiera.
CE	Borra una entrada errónea, sin destruir las operaciones pendientes.
CLR	Borra el display y todas las operaciones pendientes, aunque el número guardado en memoria se conserva.
CLM	Borra la memoria.
SQR	Extrae la raíz cuadrada del número que aparece en el display, conservando las operaciones pendientes.
X ↑ 2	Eleva el número del display al cuadrado.
:	Divide.
×	Multiplica.
−	Resta.
+	Suma.
0 al 9	Entradas numéricas.
.	Indica que las cifras siguientes son decimales.
+/−	Cambia el signo del número del display.
=	Totaliza operaciones.

Hay que tener en cuenta que el actuar sobre las cuatro operaciones básicas (+, −, ×, :) totaliza las operaciones pendientes sin tener en cuenta ninguna jerarquía matemática.

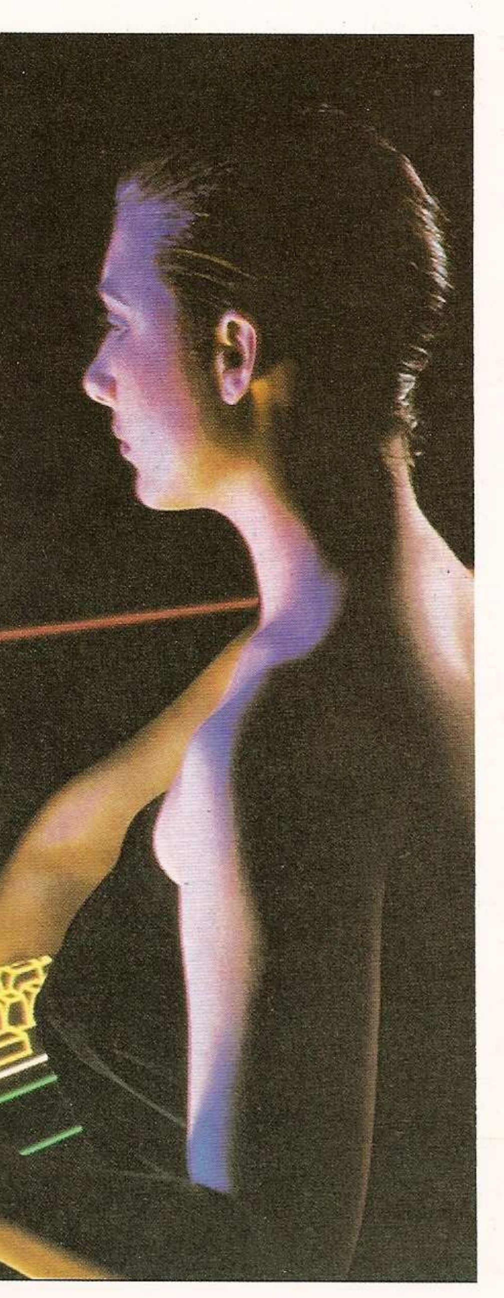
Los límites de los números manejados están fijados por la capacidad de cálculo del propio ordenador y si se sobrepasan aparece el típico mensaje de error OVERFLOW ERROR? Este límite está fijado en el número 1.70141884 E + 38.

A continuación aparece el listado del programa CALCULADORA.

Naturalmente lo primero que hace el programa es dibujar la calculadora en pantalla en color negro para así lograr una clara definición de las teclas. Hecho esto, procede a la consulta de



los valores del lápiz aprovechando el programa en código máquina que ya vimos para obtener la posición horizontal. Estos valores se comparan con unos límites para conocer la tecla que fue seleccionada, en concreto TY indica la fila en la que está situada y TX la columna. Si ambos valores son nulos se concluye que no se ha detectado ninguna tecla. Todo este proceso se desencadena si la tecla SHIFT ha sido pulsada. Conocidos el valor de TX y TY se sabe cuál es la función a realizar, pero antes de acceder a ella se pone la tecla en rojo. Debido a que las funciones están en subrutinas es muy fácil modificarlas para que una tecla concreta ejecute una operación distinta de la prevista inicialmente.



La variable principal del programa es AC, ya que en ella se almacena el resultado de los cálculos. NA es una variable intermedia en la que se va guardando el número entrado; sobre ella actúan las funciones SQR y X ↑ 2. Se manejan también algunos indicadores o «flag», como YA, indicativa de que se ha de totalizar la operación pendiente, y DP, que expresa que estamos en modo decimal.

DIBUJANDO CON EL LAPIZ EN ALTA RESOLUCION

Ésta es quizá la aplicación más evidente e interesante a la que puede ir destinado nuestro lápiz. Parece claro que un lápiz, sea óptico o no, debe servir para dibujar, pues bien, con el siguiente programa podremos utilizar la pantalla del televisor como si fuera una hoja de papel y realizar en ella vistosos dibujos utilizando el lápiz óptico. Para evitar que el programa se haga demasiado complejo y lento es necesario utilizar el cartucho SIMON'S BASIC, ya que dispone de una potencia gráfica mucho mayor que el BASIC original del Commodore, aunque también es posible escribir un programa para dibujar con el lápiz sin utilizar dicho cartucho. Aparte de la función básica de dibujar punto a punto, se incluyen otras funciones para ayudar en la creación de los dibujos. Estas funciones son las siguientes:

—LINE: Traza una línea recta entre dos puntos.

—CIRCLE: Dibuja una circunferencia tomando como datos el centro de la misma y su radio.

—FILL: Rellena una área delimitada por una línea cerrada con un color determinado.

También existe además la función PLOT, que va dibujando puntos por donde pasa el lápiz.

```
5 REM****DISEÑO DE LA
  FLECHA****
1Ø DESIGN Ø,49152+2Ø48
2Ø @.......B.............
3Ø @......BB............
4Ø @.....BBB............
5Ø @...BBBBB............
6Ø @.BBBBBBBBBBB........
7Ø @BBBBBBBBBBBB........
8Ø @.BBBBBBBBBBB........
9Ø @...BBBBB............
1ØØ @.....BBB............
11Ø @......BB............
12Ø @.......B............
13Ø @....................
14Ø @....................
15Ø @....................
16Ø @....................
17Ø @....................
18Ø @....................
19Ø @....................
2ØØ @....................
21Ø @....................
22Ø @....................
23Ø INPUT
    "[SHIFT+CLR/HOME][
    2*CRSR ABAJO][ 3*CRSR
    DCHA.]COLORES DE DIBUJO:
    C1,C2",C1,C2:
    COLOUR 5,Ø
24Ø HIRES 5,6:MULTI 1,
    C1,C2
25Ø MOB SET Ø,32,1,Ø,Ø
26Ø MOB SET 1,32,2,Ø,Ø
27Ø TEXTØ,Ø,"[1][CTRL+9]PLO
    T[CTRL+Ø][COMM+ +]LIN
    E[COMM+ +]CIRC[COMM
    + +]FILL[COMM+ +]",1,
    1,8
28Ø MO=1:CO=1:RLOCMOB Ø,
    17Ø,145,3,1
285 REM****OBTENCION
    COORDENADAS****
29Ø NX=Ø:FOR I=1 TO
    16:NX=NX+(PEEK(53267)-
    6Ø):NEXT :NX=NX/16:NY=(
    PEEK(53268)-5Ø)
3ØØ IF NX<Ø OR NY<Ø THEN29Ø
31Ø SX=NX*2+25:SY+NY+38:
    RLOCMOB Ø,SX,SY,3,1
311 REM****CAMBIO DE
    COLOR****
312 IF PEEK(197)=57
    THENCO=1
314 IF PEEK(197)=56
    THENCO=2
316 IF PEEK(197)=59
```

```
THENCO=3
320 IF PEEK(653)=Ø
    THENSP=Ø:GOTO 29Ø
330 IF SY<5Ø THEN41Ø
335 REM****EJECUCION DE LAS
    FUNCIONES****
340 ON MOGOTO 35Ø,36Ø,38Ø,
    4ØØ
350 IF SP=Ø THENSP=1:CX=
    NX:CY=NY:PLOT CX,CY,CO:
    GOTO 29Ø
355 LINE CX,CY,NX,NY,
    CO:CX=NX:CY=NY:GOTO
    29Ø
360 IF YA=1 THENLINE NX,NY,
    CX,CY,CO:RLOCMOB 1,Ø,Ø,
    Ø,1:YA=Ø:GOTO 29Ø
370 YA=1:CX=NX:CY=NY:MMOB
    1,Ø,Ø,SX,SY,3,1:GOTO 29Ø
380 IF YA=Ø THEN37Ø
390 R=SQR((CX-NX)12+(CY-
    NY)↑2/4)
395 CIRCLE CX,CY,R,R*2,
    CO:RLOCMOB 1,Ø,Ø,Ø,
    1:YA=Ø:GOTO 29Ø
400 PAINT NX,NY,CO:GOTO 29Ø
405 REM****CAMBIO DE
    FUNCION****
410 FOR I=Ø TO 8:LINE Ø,I,16Ø,
    I,Ø:NEXT
420 ON INT (NX/4Ø)+1 GOTO
    43Ø,44Ø,45Ø,46Ø
430 TEXT Ø,Ø,"[1][CTRL+9]PL
    OT[CTRL+Ø][COMM+ +]LI
    NE[COMM+ +]CIRC[COM
    M+ +]FILL[COMM+ +]",
    1,1,8:MO=1:GOTO 29Ø
440 TEXT Ø,Ø,"[1]PLOT[COMM
    + +][CTRL+9]LINE[CTR
    L+Ø][COMM+ +]CIRC[CO
    MM+ +]FILL[COMM+ +
    ]",1,1,8:MO=2:GOTO
    29Ø
450 TEXT Ø,Ø,"[1]PLOT[COM
    M+ +]LINE[COMM+ +][C
    TRL+9][CIRC[CTRL+Ø][CO
    MM+ +]FILL[COMM+ +
    ]",1,1,8:MO=3:GOTO 29Ø
460 TEXT Ø,Ø,"[1]PLOT[COMM
    + +]LINE[COMM+ +]CIR
    C[COMM+ +][CTRL+9]FIL
    L[CTRL+Ø][COMM+ +]",1,
    1,8:MO=4:GOTO 29Ø
```

El programa es muy sencillo de utilizar. Al ponerlo en marcha, nos aparece una flecha blanca que apunta a la coordenada de dibujo. Situando el lápiz sobre ella, es posible desplazarla en cualquier dirección. El desplazamiento debe efectuarse de forma lenta para que éste siga bien al lápiz. Esto se debe a que es necesario hacer una serie de medias estadísticas para obtener una coordenada horizontal estable, como ya hemos visto. Inicialmente el programa trabaja con la función PLOT, pero para cambiarla basta con desplazar la flecha hacia la parte superior de la pantalla donde aparecen representadas las posibles funciones. Seleccionando una de ellas con la flecha y pulsando la tecla SHIFT de la izquierda se cambia el modo de funcionamiento y se pasa al deseado. En todo momento, el modo en el que estamos aparece representado en vídeo inverso. A continuación se detalla el funcionamiento de cada función:

—PLOT: Aparece un punto en el lugar que indica la flecha al apretar la tecla SHIFT izquierda. El trazo que se obtiene es continuo si la mantenemos apretada, bien sea manualmente o con SHIFT LOCK.

—LINE: Para trazar una línea recta basta con llevar la flecha blanca hasta el origen de la misma y apretar SHIFT. Aparecerá una flecha roja sobre dicho punto para recordárnoslo. Llevando ahora la flecha blanca al otro extremo de la recta y apretando SHIFT de nuevo, la recta quedará trazada, desapareciendo la flecha roja.

—CIRCLE: Estando seleccionada esta función, lo primero que debemos hacer es seleccionar el centro de la circunferencia, de la manera habitual. Aparecerá como siempre la flecha roja en dicho punto. Seleccionando ahora un punto por donde deba pasar la circunferencia, ésta se trazará automáticamente.

—FILL: Para rellenar un área basta con seleccionar un punto interior a la misma con la flecha blanca y pulsar a continuación la tecla SHIFT.

El programa permite, además, trabajar en modo multicolor, ya que es posible utilizar cuatro colores simultáneamente en la pantalla. Dos de esos colores ya están asignados, y son concretamente el negro para el fondo de la pantalla y el blanco. Los otros dos pueden ser elegidos por el usuario al principio del programa. Antes de empezar a dibujar, el programa pide dichos colores que deben introducirse con los números habituales asignados por Commodore. Se utiliza el blanco como un color fijo para ganar en velocidad. Si situamos el lápiz sobre una zona blanca la flecha se desplazará instantáneamente a esa zona. Así la selección de funciones podrá hacerse con rapidez. La forma de elegir los colores de dibujo es la siguiente: pulsando la tecla ← el color de trabajo será el blanco; oprimiendo «1» pasaremos a utilizar el primer color preseleccionado; y haciendo lo propio con «2» se utilizará el segundo.

El programa desarrollado utiliza comandos propios del cartucho SIMON'S BASIC, por lo que puede resultar un tanto extraño. Lo primero que hace es diseñar la flecha que se moverá por la pantalla. A continuación se piden los colores de dibujo y se pasa a la pantalla de alta resolución en la que se va a trabajar. Se leen entonces los valores del lápiz desplazando la flecha según lo que éstos indiquen. Así es posible mover la flecha por cualquier lugar de la pantalla. En el caso de que se haya pulsado la tecla SHIFT se sale del bucle de movimiento y comprueba si se ha efectuado alguna nueva selección. En tal caso se cambia la bandera de selecciones (MO) y se pone en vídeo inverso la función elegida. En caso de haber pulsado SHIFT y no haber hecho ninguna selección el programa pasa a ejecutar la función actual. En caso de PLOT,

se irá trazando una línea que una el punto anterior con el actual mientras no deje de pulsarse SHIFT. Las otras funciones se realizan directamente utilizando comandos de la ampliación del BASIC. Durante el bucle de movimiento también se comprueba si alguna de las teclas de color ha sido pulsada, pasándose en tal caso a actualizar el valor de la variable CO que contiene el número de color empleado en el dibujo.

Existe un compromiso entre la velocidad del movimiento de la flecha y la precisión del mismo. La necesidad de hacer una media aritmética provoca la ralentización del bucle de movimiento. Podemos acelerar el movimiento de la flecha si tomamos un número menor de muestras; no obstante esto traerá consigo un empeoramiento de la precisión horizontal. Por contra lograremos aumentarla tomando un número mayor de ellas, descendiendo lógicamente la velocidad de desplazamiento de la flecha. Es aconsejable evitar dibujar cerca de los bordes, donde la precisión es menor como ya se ha comentado más arriba.

El programa puede ampliarse según las necesidades de cada uno. Por ejemplo, puede realizarse un volcado de la pantalla de alta resolución en la impresora mediante las instrucciones que incluye el SIMON'S para ello.

ALGUNA MEJORA EN EL LAPIZ OPTICO

En algunos casos puede resultar incómodo el tener que utilizar el teclado para validar la posición del lápiz. Una forma de independizar el lápiz del ordenador es añadirle al lápiz un pulsador solidario al mismo en una posición cómoda para el usuario. Este pulsador sustituiría la tecla SHIFT en los programas descritos anteriormente. Será, entonces, necesario modificar una sola instrucción de los mismos. Concretamente deberá sustituirse las intrucciones

XXX WAIT 653,1

por

XXX IF PEEK(56321)AND1 THEN XXX

o

XXX IF PEEK(653)=∅ THEN YYY

por

XXX IF PEEK(56321)AND1 THEN YYY

donde XXX y YYY son las líneas de programa.

El interruptor solidario con el lápiz deberá ir conectado a los pins del port 1 números 1, correspondiente a una de las entradas del joystick, y 8, que es la destinada a masa.

Ahora podremos controlar los programas con una sola mano, ya que el efecto que antes tenía la pulsación de la tecla SHIFT lo ejerce ahora el pulsador del lápiz. Esto resulta ser en la práctica mucho más cómodo y rápido de lo que parece ser a primera vista.

Finalmente, te informamos que todos los componentes utilizados en la realización del lápiz pueden adquirirse fácilmente en los comercios especializados de componentes electrónicos. El precio aproximado del material necesario es de unas 1.200 pesetas (o al menos esto es lo que costó el prototipo).

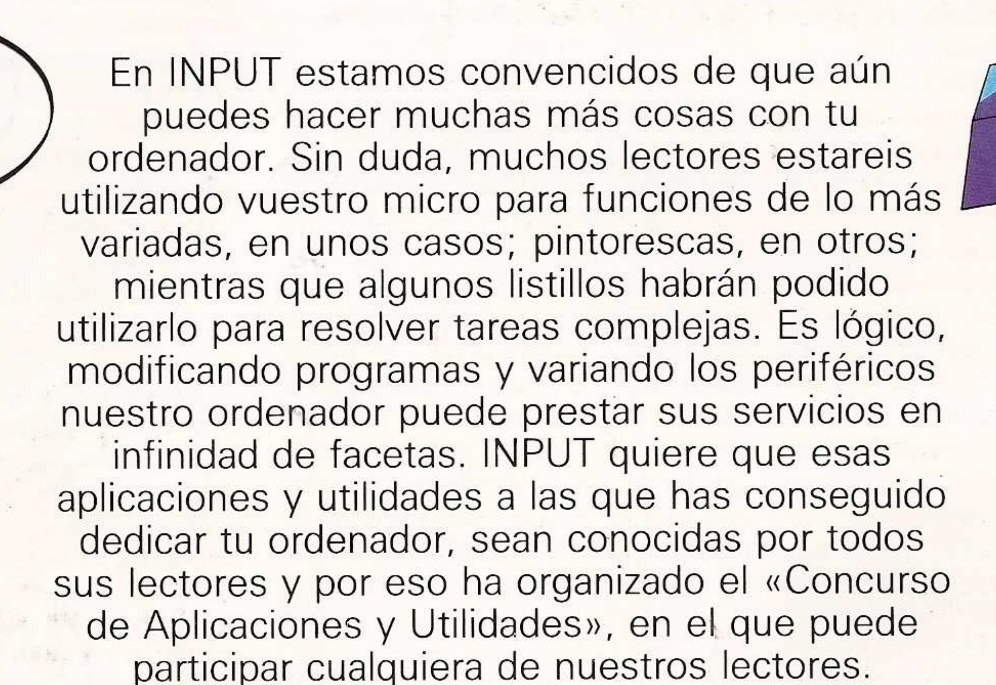
RAMÓN OLLÉ y RAMÓN SOLÉ

GANADORES DE LOS MEJORES DE INPUT COMMODORE

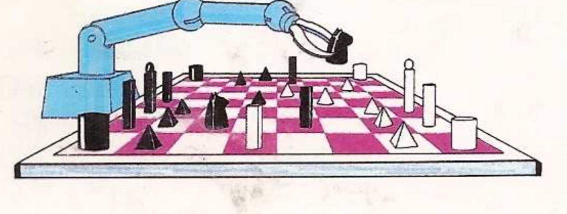
En el sorteo correspondiente al número 20 entre quienes escribisteis mandando vuestros votos a los MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Carlos Vázquez Veiga	La Coruña	Two on two
J. José Montes Huertas	Barcelona	Two on two
Mariano Arrese González	Madrid	Two on two
Fco. Javier Ros Muñoz	Barcelona	Gauntlet
Manuel Martín Giménez	Almería	Golpe en la pequeña China
Miquel Soteres Roig	Tàrrega (Lleida)	Howard
Antonio Pérez Cano	Palencia	Golpe en la pequeña China
Víctor Martínez Conte	Zaragoza	Laberinto
Santiago Rodríguez Vivas	Barcelona	Aliens
Fco. Javier de las Heras Ángel	Valencia	Aliens





En INPUT estamos convencidos de que aún puedes hacer muchas más cosas con tu ordenador. Sin duda, muchos lectores estareis utilizando vuestro micro para funciones de lo más variadas, en unos casos; pintorescas, en otros; mientras que algunos listillos habrán podido utilizarlo para resolver tareas complejas. Es lógico, modificando programas y variando los periféricos nuestro ordenador puede prestar sus servicios en infinidad de facetas. INPUT quiere que esas aplicaciones y utilidades a las que has conseguido dedicar tu ordenador, sean conocidas por todos sus lectores y por eso ha organizado el «Concurso de Aplicaciones y Utilidades», en el que puede participar cualquiera de nuestros lectores.





BASES

UTILIDADES Y APLICACIONES: Si tu ordenador controla la calefacción de tu casa, gobierna un robot, dirige un pequeño negocio, organiza la maqueta de tu tren eléctrico, o cualquier cosa interesante u original; envíanos información gráfica y listados de tus programas, grabados en un cassette, diskette o microdrive.

Todo ello habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvideis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante los próximos tres meses SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
en material microinformático a elegir por los afortunados.

¡No os desanimeis!, por muy simples o complejas que puedan parecer vuestras ideas, todas están revisadas con el máximo interés.

INPUT COMMODORE
Aribau, 185. Planta 1.ª
08021 BARCELONA

NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

EL ZORRO Y LAS OCAS (II)

- COMO PROGRAMAR TRABAJOS
- INICIALIZACION
- EJECUTANDO EL PROGRAMA
- TRAZANDO MOVIMIENTOS
- ¿OTRA PARTIDA?

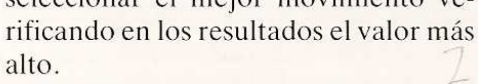
Haz uso de la última parte de la teoría para empezar a escribir el juego del Zorro y las Ocas.

He aquí la rutina para inicializar el juego y para trazar movimientos.

Esta vez vas a introducir rutinas de inicialización y la rutina principal de trazado. Asimismo, también podrás proponer al jugador otra partida, pero en este estadio aún no podrás ejecutar con RUN el programa, pues aún hay varias rutinas importantes que añadir.

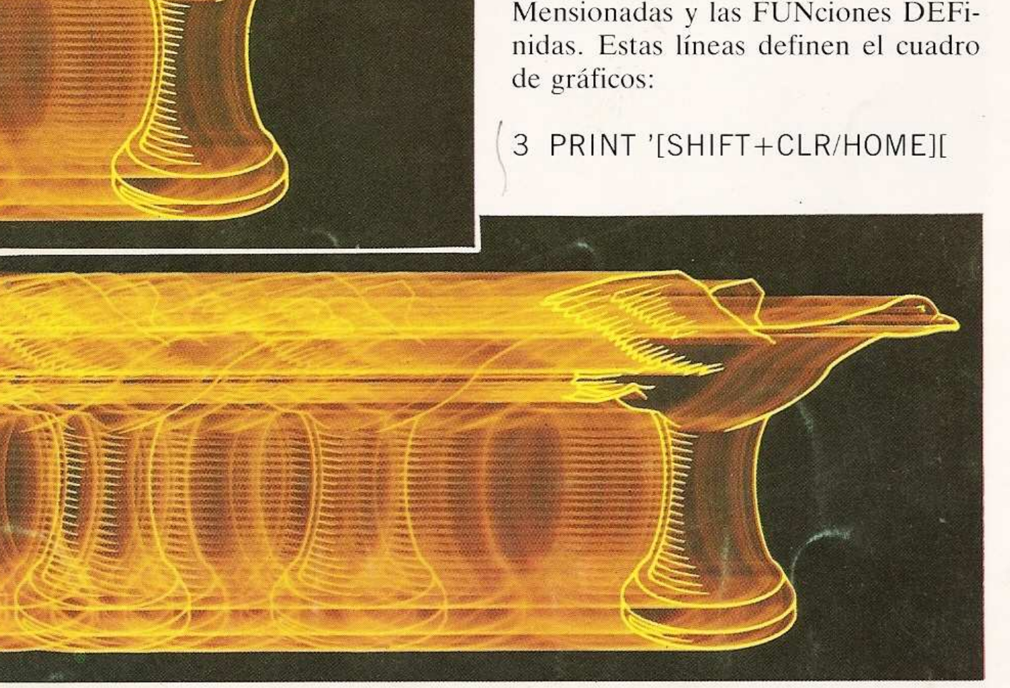
VISION GLOBAL

El programa trabaja evaluando cada una de las posiciones del juego de acuerdo con la configuración de las fichas. A cada posición se le adjudica un valor numérico en el programa; de este modo, el programa es capaz de seleccionar el mejor movimiento verificando en los resultados el valor más alto.



El programa opera en tres niveles cuando planea la jugada. El nivel uno, sólo proyecta un movimiento —se trata del así denominado *un movimiento*—. En el nivel superior del juego utiliza el algoritmo alfa-beta para ahorrar tiempo al buscar entre las muchas vías de posibilidades que progresivamente se van multiplicando. En niveles intermedios, el programa repasa todas las posibilidades existentes.

Las rutinas desde la línea 2010 hasta la línea 3000 sólo se ejecutan una vez, de ahí que estén ubicadas al final del programa. Con estas rutinas *de poco uso* situadas al final del programa, las rutinas principales pueden ser ubicadas cerca del principio del programa para lograr mayor velocidad.



INICIALIZACION

He aquí las rutinas para inicializar el juego. Las direcciones han sido DIMensionadas y las FUNciones DEFinidas. Estas líneas definen el cuadro de gráficos:

```
3 PRINT '[SHIFT+CLR/HOME][
  3*CRSR abajo]';CHR$(8)
4 x=16:FOR Z=Ø TO 15:POKE
  646,z:IF z=6 THENNEXT z
5 x=x - 1:POKE 646;z:PRINT
  TAB(5 + x)'[SHIFT+n] zorro y
  ocas [SHIFT+n]':NEXT z 1Ø
  POKE 53272;19:GOTO
  2Ø1Ø
2Ø1Ø DIM g(4)
2Ø2Ø DIM b(31):b(Ø)=1:FOR
     i=1 TO 31:b(i)=b(i - 1) *
     2:NEXT i
2Ø26 bx=b(31) * 2 -
     b(24):e=1e3Ø:h= - 1e3Ø
2Ø3Ø l2=LOG(2):DEF FN
     a(f)=INT (LOG[f] / 12 +
     .ØØ1)
2Ø4Ø DIM h$(15,1):DIM g$(1
     ):g$(Ø)='[SHIFT+l][SHIF
     T+@][ 2 espacios]":g$(
     1)=."[SHIFT+1][SHIFT+
     @][SHIFT+u][SHIFT+v]"
2Ø45 DIM h$(1):h$(Ø)='[ 2 es
     pacios][SHIFT+l][SHIFT
     +@]':h$(1)='[SHIFT+u][
     SHIFT+v][SHIFT+l][SHIF
     T+@]'
2Ø5Ø x =Ø:FOR a=Ø TO 1:FOR
     b=Ø TO 1:FOR c=Ø TO
     1:FOR d=Ø TO 1:b$(x,
     Ø)=g$(d) + g$(c) + g$(b)
     + g$(a)
2Ø6Ø b$(x,1)=h$(a) + h$(b) +
     h$(c) + h$(d):x=x +
     1:NEXT d,c,b,a
2Ø7Ø DIM s$(7):FOR aØ TO 6
     STEP2:FOR b=a * 4 TO a *
     4 + 3
2Ø8Ø s$(a)=s$(a) +
     '[SHIFT+o][SHIFT+p]' +
     RIGHT$(STR$[b],2)
2Ø81 s$(a + 1)=RIGHT$(STR$(b
     + 4),2) + '[SHIFT+o]
     [SHIFT + p]' + s$(a + 1)
2Ø82 NEXT b,a
```

```
2090 DIM f$(1):f$(0)="[SHIFT
     +I][SHIFT+@][SHIFT+w]
     [SHIFT+x]":f$(1)="[SHIF
     T+w][SHIFT+x][SHIFT+I
     ][SHIFT+@]'
2095 DEF FN c(b)=(3 AND b) *
     (4 - 2 * (4 AND b)) + 3 *
     (4 AND b)
```

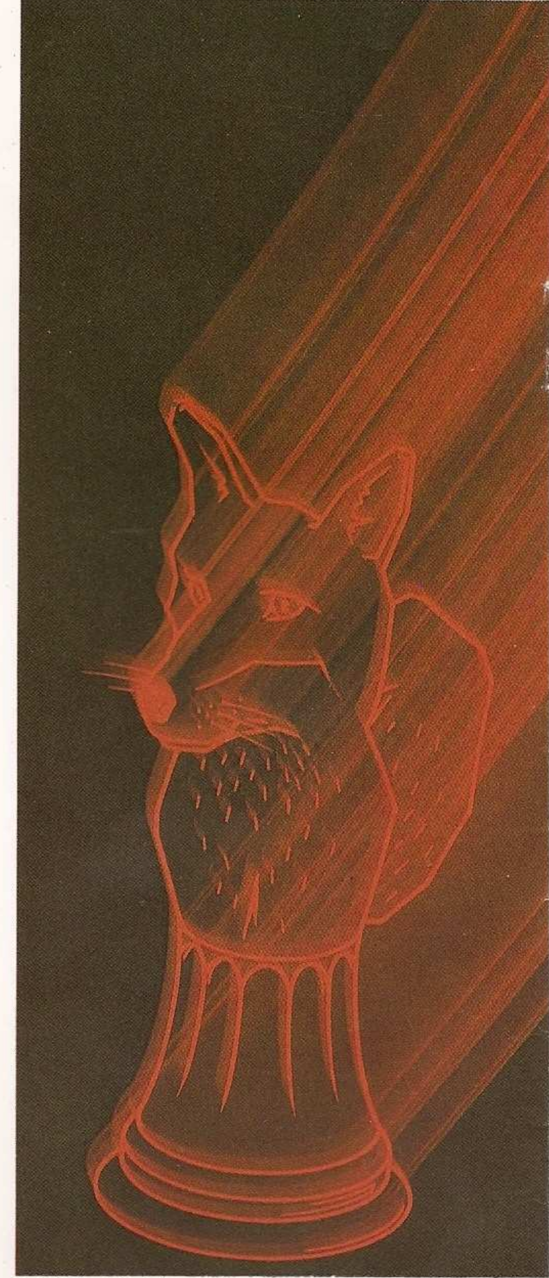
La línea 2010 DIMensiona la dirección utilizada para almacenar las posiciones de las ocas. La línea 2020 numera cada casilla del tablero que se utilizará en el juego.

La línea 2030 determina el número de configuraciones que pueden ser evaluadas por el programa. 0.001 ha sido añadido cuando se ha DEFinido la FUNción A para evitar el redondear errores cuando se utilizan LOGaritmos. La dirección B$, DIMensionada en la línea 2040, se utiliza para mostrar líneas del tablero que están ocupadas por piezas. F$ se utiliza para la pieza que representa al zorro y a la casilla blanca, y H$ para las ocas y las casillas blancas. S$ se utiliza para determinar los números de las casillas.

AL INICIO

Esta rutina permite al jugador elegir quién juega, qué piezas (si el zorro o las ocas) y seleccionar el nivel de habilidad del ordenador. ¡No es demasiado difícil —al principio—!

```
2500 DIM r(1999),s(1999)
2700 f=1:g(1)=28:g(2)=29:g(
     3)=30:g(4)=31:GOSUB
     2710:GOTO1010
2710 PRINT '[SHIFT+CLR/
     HOME]quieres ser el zorro
     (s/n)?'
2720 GET i$:pf=0:IF i$='s'
     THEN2760
2730 pf=1:IF i$<>'n'
     THEN2720
2740 $f:INPUT '[SHIFT+CLR/
     HOME]nivel de juego del
     zorro (1-10)';sf:IF sf<1
     ORsf>10
     THEN2740
2750 hf= - 131 * (sf=5) - 613
     * (sf=6) - 1997 * (sf>6)
2760 PRINT '[SHIFT+CLR/
     HOME]quieres ser las ocas
     (s/n)?'
2770 GET i$:pg=0:IF i$='s'
     THEN2860
2780 pg=1:IF i$<>'n'
     THEN2770
2790 sg=0:INPUT '[SHIFT+CLR/
     HOME]nivel de juego de las
     ocas (1-10)';sg:IF sg<1 OR
     sg>10 THEN2790
2800 hg= - 131 * (sg=5) -
     613 * (sg=6) - 1997 *
     (sg>b):IF hf<hg
     THENhf=hg
2860 PRINT '[SHIFT+CLR/
     HOME]deseas modificar la
     posición inicial'
2870 PRINT '[ 13*CRSR
     dcha.](s/n) ?'
2875 GET i$:IF i$='n'
     THEN3000
2880 IF i$<>'s' THEN2875
2890 GOSUB 210:GOSUB
     310:PRINT 'quieres mover
     el zorro (s/n)? ';
2900 GET i$:IF i$='n'
     THEN2930
2903 IF i$<>'s' THEN2900
2915 PRINT 's'
2920 INPUT '[ 8*CRSR
     dcha.]mover zorro a';f:IF
     f<0 OR f>31 THEN2920
2925 GOSUB 340
2930 FOR g=1 TO 4:GOSUB
     210:GOSUB 310
2940 PRINT 'quieres mover la
     oca n.';g(g):PRINT '(s/n)? ';
2950 GET i$:IF i$='n'
     THEN2990
2960 IF i$<>'s' THEN2950
2965 PRINT 's'
2970 INPUT '[ 8*CRSR
     dcha.]mover oca a';
     i:GOSUB 340
2971 IF FN x(i) OR i=f
     THENPRINT TAB(8);'casilla
     ocupada':GOTO 2940
2972 IF i<0 OR i>31
     THEN2970
2980 g(g)=i
```



```
2990 NEXT g:IF FN x(f)
     THENPRINT TAB(8);'hay
     una oca debajo del zorro'
2995 FOR i=1 TO 1500:NEXT i
3000 RETURN
```

La línea 2700 determina la posición de partida, con las cuatro ocas ocupando las cuatro casillas de la parte inferior del tablero, y el zorro ocupando la segunda casilla de la línea superior izquierda del tablero.

Una vez que la línea 2700 ha inicializado la posición de partida del zorro y las ocas, las líneas 2710 a 2750 dan al jugador la opción de jugar a ser el zorro y determinan quién va a jugar a ser el zorro. Las líneas 2760 a 2800 son



similares, sólo que aquí al jugador se le da la opción de jugar a ser las ocas.

El juego ha sido diseñado de manera que permita, o bien ajustar la posición de partida, o bien continuar en la posición de partida en la que dejaste el juego la última vez (necesitarás tomar notas de las posiciones de las fichas cuando acabe el juego) o bien intentar ganar (¡o perder!) desde una posición particularmente interesante. Las líneas 2860 a 3000 preguntan si el jugador desea modificar la posición de partida, dan la señal de *preparado* y aseguran que las posiciones elegidas sean correctas.

TRAZANDO MOVIMIENTOS

La rutina de trazado de movimientos es una de las más importantes del programa.

```
21ØØ DIM r$(7)
211Ø DEF FN f(b)=((b>3) +
     (b<28)) * (((3 AND b)<3)
     - 1) - 1
212Ø DEF FN g(b)=(b>3) * (((3
     AND b)<3) - 1) - 1
213Ø DEF FN m(a)=b - 2 * (3
     AND b) - 2 - 8 * (b<4
     OR a>1) - (1 + a * 7) *
     ((3 AND b)=3) + (1 AND
     a)
214Ø DEF FN x(b)=(b=g(1) OR
     b=g(2) OR b=g(3) OR
     b=g(4))
2142 DEF FN z(b)= - (b=g(1))
     - (b=g(2)) * 2 - (b=g(3))
     * 3 - (b=g(4)) * 4
215Ø DIM m(3,31),x(31),z(31)
216Ø FOR b=Ø TO 31:FOR a=Ø
     TO 3:m(a,b):FN m(a):NEXT
     a:x(b)=FN f(b):z(b)=FN
     g(b):NEXT h
```

Las líneas 2110 a 2160 trazan el plan de movimientos del zorro y de las ocas en la dirección M. A lo largo de este plan también, se ajustan el número de posibles movimientos del zorro, dirección X, así como el número de posibles movimientos de las ocas, dirección Z. Las direcciones son copias de las funciones en las líneas 2110 a 2142.

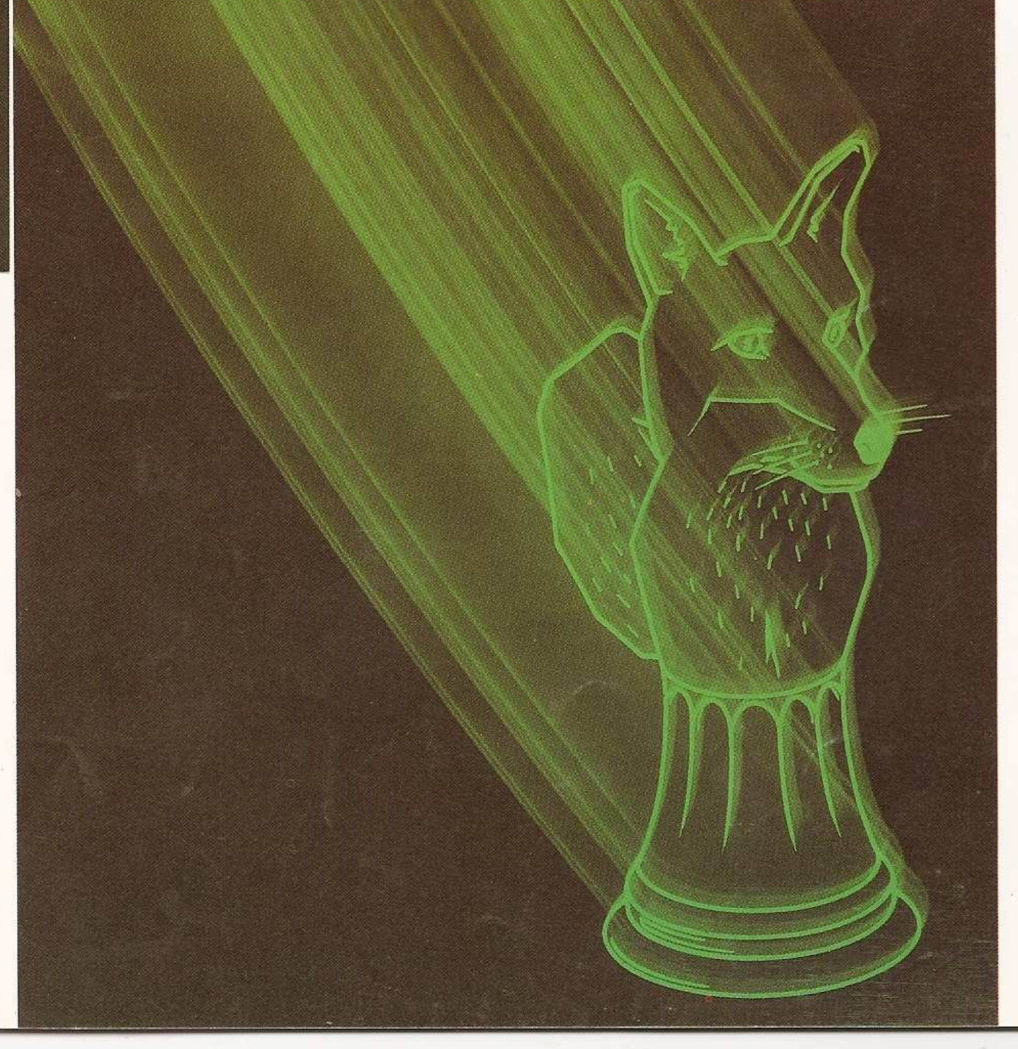
¿OTRA PARTIDA?

Añade a continuación otra rutina «¿Otra partida?»

```
141Ø PRINT TAB(8);'deseas
     jugar otra vez (s/n)?'
142Ø GET i$:IF i$='s'
     THEN27ØØ
143Ø IF i$<>'n' THEN142Ø
144Ø PRINT '[SHIFT+CLR/
     HOME][CTRL+2]':POKE
     53272,21:END
```

Estas líneas deberían resultarnos ya familiares, y entran en acción cuando las ocas intentan atrapar al zorro, o cuando el zorro intenta alcanzar el lado opuesto del tablero.

No intentes ejecutar el programa con RUN en esta fase, pues aún hay que añadirle muchas partes de vital importancia. En la siguiente parte del artículo, insertarás las rutinas que te permitirán ejecutar el juego.



EL ZORRO Y LAS OCAS (Y III)

Deja que tu ordenador recoja ideas antes de comenzar el juego. Con estas rutinas podrás jugar el papel del zorro o de las ocas y adelantarte para improvisar su jugada.

Ésta es la tercera y última parte del juego del Zorro y las Ocas. Las rutinas que quedan son las que permitirán al ordenador jugar la parte, ya sea del zorro o de las ocas. De hecho se puede programar al ordenador para que juegue ambas partes a la vez, cuando compita contra sí mismo, o bien puedes programarlo para que no juegue a ser ninguna de las figuras, y jugar tú contra algún amigo utilizando el ordenador en lugar de un tablero y fichas.

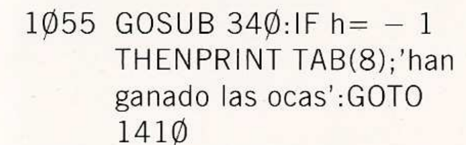
```
21Ø  p=b(g(1)) + b(g(2)) +
     b(g(3)) + b(9(4))
22Ø  x=f:IF p<b(31) THEN p=p
     - hx:x=31 - f
23Ø  p=p * h(x):RETURN
25Ø  f=FN a(ABS(p)) - 31 :b=p /
     b(f):IF b<Ø THENb=b +
     bx:f=31 - f
26Ø  FOR a=1 TO 4:9(a)=FN
     a(b):b=b - b(9(a)): NEXT
     a:RETURN
```

Estas son dos de las rutinas más importantes del programa. La rutina que comienza en la línea 210 hasta la línea 230 evalúa la posición de juego resumiéndola en un número de una sola cifra. Luego, una vez que el programa ha decidido el valor del mejor movimiento, ha de convertir ese número en un movimiento o en un juego de posiciones para las fichas. Las líneas 250 a 260 transforman el valor simple en el juego de valores necesarios para posicionar las fichas.

Estas subrutinas son llamadas con frecuencia durante el programa, de manera que deberán estar colocadas correctamente al inicio del mismo.

EL MOVIMIENTO DEL ZORRO

```
1Ø1Ø  GOSUB 21Ø
1Ø2Ø  GOSUB 31Ø:GOSUB 25Ø
1Ø3Ø  IF f>27 THENPRINT
      TAB(8);'ha ganado el
      zorro':GOTO 141Ø
1Ø32  GOSUB 41Ø:IF v=h
      THENPRINT TAB(8);'han
      ganado las ocas':GOTO
      141Ø
1Ø4Ø  IF pf THENPRINT TAB(8);
      'estoy pensando...':GOTO
      111Ø
1Ø5Ø  INPUT '[ 8*CRSR
      dcha.]mover el zorro a';b
```

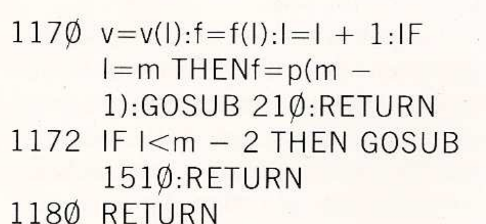


```
1Ø55  GOSUB 34Ø:IF h= - 1
      THENPRINT TAB(8);'han
      ganado las ocas':GOTO
      141Ø
1Ø6Ø  FOR a=Ø TO x(f):x=m(a,
      f):IF x=b THEN IF NOT FN
      x(x) THENf=bGOSUB
      21Ø:GOTO 12ØØ
1Ø7Ø  NEXT a:PRINT TAB(8);
      'movimiento ilegal':GOTO
      1Ø5Ø
111Ø  l=sf:m=sf:v(m)=e * m:IF
      m>4 THENFOR a=Ø TO
      hf:r(a)=Ø:NEXT a
1112  GOSUB 112Ø:GOTO 12ØØ
112Ø  IF l=1 THEN41Ø
1122  IF l<m - 2 THENGOSUB
      161Ø:IF V<>Ø
      THENRETURN
```



```
113Ø  l=l - 1:v(l)=h *
      l:a(l)=x(f):f(l)=f
114Ø  f=m(a(l),f(l))
115Ø  IF FN x(f)<>Ø
      THEN116Ø
1155  GOSUB 132Ø:IF v>v(l)
      THENv(l)=v:(l)=f:IF v>v(l
      + 1) THEN f=f(l):l=l +
      1:RETURN
116Ø  a(l) = a(l) - 1:IF a(l)>=Ø
      THEN114Ø
```

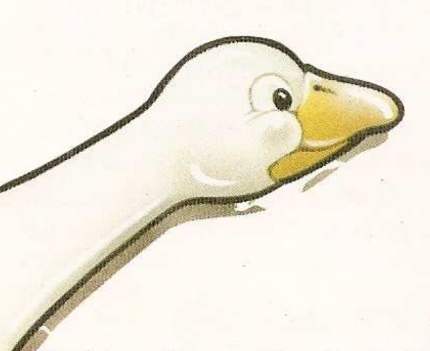





```
117Ø v=v(l):f=f(l):l=l + 1:IF
     l=m THENf=p(m −
     1):GOSUB 21Ø:RETURN
1172 IF l<m − 2 THEN GOSUB
     151Ø:RETURN
118Ø RETURN
```

Las líneas 1020 a 1180 dirigen el movimiento del zorro. La rutina utiliza la rutina de visualización del tablero para mostrar el estado actual del tablero, seguidamente pasa a comprobar si el zorro ha ganado en la línea 1030. En la línea 1032 comprueba si hay allí, al menos, un movimiento legal que pueda realizar el zorro (de lo contrario, ganan las ocas).





Si el jugador es quien lleva el control de las ocas, las líneas 1050 a 1070 se encargan de introducirlo en el ordenador y de comprobar que el movimiento efectuado sea efectivamente correcto. Si, por otra parte, es el ordenador el que controla al zorro, las líneas 1110 a 1112 son las encargadas de llevar a cabo el movimiento. El número de pliegues que el programa observa, M, y el pliegue actual considerado, L, son colocados en la línea 1110. Las líneas 1120 a 1180 son una subrutina que evalúa el mejor movimiento.

Se utilizan cuatro direcciones en la rutina para el mejor movimiento: A contiene el número de movimientos que aún pueden ser intentados; V, el mejor resultado hasta el momento; P, el movimiento que ha conseguido este resultado, y F, la posición previa del zorro.

MOVIENDO LAS OCAS

```
5ØØ  v= − 1:FOR c=1 TO
     4:g=g(c): IF z(g)<Ø
     THENNEXT :RETURN
5Ø2  FOR a=Ø TO z(g):x=m(a,
     g):v=v AND (FN x(x) OR
     x=f):NEXT :NEXT :RETURN
12ØØ GOSUB 31Ø:GOSUB 25Ø
12Ø2 IF f>27 THENPRINT
     TAB(8);'ha ganado el
     zorro':GOTO 141Ø
12Ø4 GOSUB 5ØØ:IF v
     THENPRINT TAB(8);'ha
     ganado el zorro':GOTO
     141Ø
121Ø IF pg THENPRINT TAB(8);
     'estoy pensando...':GOTO
     131Ø
122Ø INPUT '[ 8*CRSR
     dcha.]que oca vas a mover';
     g
1225 GOSUB 34Ø
123Ø c=FN z(g):IF c=Ø
     THENPRINT TAB (8);'no
     hay ocas en";9:GOTO 122Ø
124Ø INPUT '[ 8*CRSR
     dcha.]hasta donde';i
1245 GOSUB 34Ø:IF i= − 1
     THENPRINT TAB(8);'el
     zorro ha ganado':GOTO
     141Ø
125Ø IF FN x(i) THENPRINT
     TAB(8);'no encima de otra
     oca':GOTO 122Ø
126Ø IF i=f THENPRINT TAB(8);
     'no encima del zorro':GOTO
     122Ø
127Ø FOR a= − 1TO z(g):IF
     a>=Ø THENIF m(a,g)=i
     THEN g(c)=i:GOTO 1Ø1Ø
128Ø NEXT a:PRINT TAB(8);
     'movimiento ilegal':GOTO
     122Ø
131Ø l=sg:m=sg:v(m)=h * m:IF
     m>4 THENFOR a=Ø TO
     hf:r(a)=Ø:NEXT a
1312 GOSUB 132Ø:GOTO 1Ø2Ø
132Ø IF l=1 THEN51Ø
1322 IF l<m − 2 THENGOSUB
     161Ø:IF v<>Ø
     THENRETURN
1324 l=l − 1:v(l)=e * l:c=1
133Ø c(l)=c:f(l)=g(c):a(l)=Ø:IF
     a(l)>z(g(c)) THEN1362
134Ø b=m(a(l);f(l)):x=FN
     x(b):g(c)=b : IF x OR b=f
```

```
      THEN136Ø
.35Ø  GOSUB 112Ø:c=c(l)
.355  IF v<v(l)
      THENv(l)=v:p(l)=g(c) + c
      * 32:IF v<v(l +
      1)THENg(c)=f(l):l=l +
      1:RETURN
.36Ø  a(l)=a(l) + 1:IF
      a(l)<=z(f(l)) THEN134Ø
.362  g(c)=f(l):c=c + 1:IF c<5
      THEN133Ø
.37Ø  v=v(l):l= l + 1:IF l=m
      THENc=INT (p(l − 1) /
      32):g(c)=p(l − 1) AND
      31:GOSUB21Ø:RETURN
.372  IF l<m − 2 THENGOSUB
      151Ø:RETURN
.38Ø  RETURN
```

La rutina que va de la línea 1200 a a línea 1380 dirige el movimiento de as ocas. Hay rutinas para el caso en que el jugador controla las ocas —líneas 1220 a 1290—, y para el caso en que es el ordenador quien controla las ocas —líneas 1320 a 1380.

LOS MEJORES MOVIMIENTOS

```
41Ø  v=h:FOR a=x(f) TO Ø STEP
     − 1:x=m(a,f):IF FN x(x)<Ø
     THENNEXT :l=1:RETURN
42Ø  b=f:f=x:GOSUB
     21Ø:v=p:f=b:l=1:RETURN
51Ø  v=e:FOR c=1 TO
     4:g=g(c):IF − bg>v
     THEN530
52Ø  FOR a=Ø TO z(g):x=m(a,
     g):IF FN x(x) OR x=f
     THENNEXT :GOTO 53Ø
528  v=b(x) − b(g):d=c:b=x
53Ø  NEXT
     c:g=g(d):g(d)=b:GOSUB
     21Ø:v=p:g(d)=g:RETURN
```

Estas rutinas están relacionadas con el «un movimiento», en otras palabras, en esta fase, el ordenador sólo programa un solo movimiento cuando está buscando el mejor movimiento.

Las líneas 410 a 420 repasan todos os posibles movimientos que puede realizar el zorro; utilizando la dirección M de trazado, coloca la subrutina comenzando en la línea 2110 (ver anterior). La subrutina devuelve un valor de P, configuración posterior al mejor movimiento, y V, evaluación posterior al mejor movimiento.

Las líneas 510 a 530 son una rutina similar, pero en esta ocasión, el ordenador calcula cuál es el mejor movimiento para las ocas. P y V están fijadas de igual modo que en la rutina previa.

En ambos casos, GOSUB 210 recoge el mejor movimiento de entre todos los evaluados.

LA TABLA HASH-CODE

```
151Ø  GOSUB 21Ø:c=p
152Ø  c=c − INT ((c / hf + c) − )
      c) * hf:IF c<Ø OR c>=hf
      THEN152Ø
155Ø  FOR a=c TO c + 4: IF
      r(a)<>Ø AND r(a)<>p
      THENNEXT :RETURN
156Ø  r(a)=p:s(a)=v:RETURN
161Ø  GOSUB 21Ø:c=p
162Ø  c=c − INT ((c / hf + c) −  )
      c) * hf:IF c<Ø OR c>=hf
      THEN162Ø
165Ø  FOR a=c TO c +4:IF
      r(a)<>Ø AND r(a)<>p
      THENNEXT :v=Ø:RETURN
166Ø  v= − s(a) *
      (r(a)=p):RETURN
```

En los niveles superiores de juego preferirás aplicar el algoritmo alfabeta que ya hemos visto. En efecto, ya has utilizado previamente el algoritmo como parte de las rutinas de los movimientos del zorro y del de las ocas. Antes de aplicar el algoritmo, comprueba si es conveniente utilizar el algoritmo (¿el nivel de juego elegido es suficientemente alto como para garantizar su utilización?).

La rutina que va de la línea 1110 a la línea 1150 controla al zorro mientras que la rutina que va de la línea 1310 a la línea 1350 controla las ocas.

El algoritmo se aplica en el último test IF al final de las líneas 1150 y 1350, después de que V (M) ha sido colocado al nivel apropiado en las líneas 1110 y 1310.

El algoritmo alfa-beta es más eficiente si el ordenador considera primero cuáles son los mejores movimientos —para las ocas, la casilla del número más alto en cada fila de cuatro casillas, y para el zorro, la casilla abierta para él y que esté más cercana al final de las ocas.

El algoritmo alfa-beta se utiliza conjuntamente con una técnica conocida como «*hash-coding*» con el fin de confeccionar y utilizar una tabla de los movimientos ya considerados. *Hash-coding* permite al ordenador comprobar rápidamente si el movimiento ya ha sido considerado. Cuanto más larga sea la tabla de *hash-coding* dentro del programa, más rápidamente se debería poder ejecutar el programa.

La tabla se inicializa en las líneas 2500, 2750 y 2800. Hay «mejores valores» teóricos para las dimensiones de las direcciones que sostienen las tablas —línea 2500—. Las direcciones han sido DIMensionadas lo más largas posible, indicando la memoria RAM disponible en cada ordenador.

La tabla es puesta a cero en las líneas 1110 y 1310, los contenidos se verifican en las líneas 1122 y 1322, y se colocan en las líneas 1172 y 1372.

La subrutina de verificación se inicia en la línea 1610 y la subrutina de ajuste, en la línea 1510.



FREDDY Y LA ARAÑA DE MARTE (I)

- LA TRAMA
- COMO SE PROGRAMA EL JUEGO
- OBTENCION DE LOS GRAFICOS
- FREDDY, FLECHAS, GLOBOS Y LA ARAÑA

Abandona a su destino a Freddy y a la voraz araña de Marte mientras te ocupas de reconstruir uno de los juegos electrónicos más típicos. En el siguiente artículo tendrás el programa completo.

«Freddy y la araña de Marte», o «La pesadilla del limpiador de cristales», es un juego electrónico que, para ofrecértelo completo, nos va a ocupar dos artículos, pero a cambio vas a conocer un clásico de los juegos electrónicos.

En este primer artículo se establecen los gráficos y se inicializa el programa. En el siguiente, obtendrás el programa entero con sólo coser las rutinas entre sí.

EL JUEGO

El punto de partida para diseñar un juego es, claro está, una idea: una trama o un argumento pueden servir de base para construir el juego.

En el que vamos a examinar, Freddy es un limpiacristales que tiene pánico de las arañas. Ni las consultas al médico ni a innumerables especialistas han logrado que supere su fobia. Tan grave es ésta, que ahora le da por soñar pesadillas sobre una araña marciana, enorme, horrible y voraz, o por tener sueños con globos y lances con el tiro al arco, su deporte favorito.

A menudo se ha sorprendido sudando a chorros, despierto después de haber soñado que estaba acorralado, subido a su escalera y armado no de su habitual cubo de la limpieza sino de un arco y una aljaba repleta de flechas. Toda su ansia se había concentrado en hacer explotar esos globos temibles que, si logran llegar hasta la jaula de la araña que pende sobre su cabeza, está perdido, pues abrirán las puertas y liberarán al no menos temido arácnido.

O te afanas en ayudar al pobre Freddy o su final es tan seguro como apetitoso resultará a la araña el desayuno que imaginas.

Hasta aquí el argumento o la trama. En el marcador se anotarán tantos puntos cuantos globos se hayan con-



seguido explotar. Pero las pesadillas no suelen acabar tan fácilmente: la tortura del pobre limpiador de ventanas aumenta de grado a medida que los globos pasan delante de él cada vez más veloces. Con que falles en la destrucción de tres de éstos, la araña campará por sus respetos.

COMO SE ESCRIBE EL JUEGO

Son cuatro los objetos que poseen movimiento en la pantalla: la araña, que se mueve tanto horizontal como verticalmente; el globo, que surca el espacio y que sólo se mueve verticalmente; Freddy, que también se mueve en dirección vertical únicamente, y la flecha que éste dispara. La flecha se mueve en general horizontalmente, pero cuando a Freddy le da por subir o bajar por la escalera entonces la flecha habrá de moverse verticalmente junto con nuestro personaje. Sin embargo, quienes más nos deben preocupar son la araña y el globo de turno. Son muchas las variables que se asocian a estos dos objetos. La técnica que mejor se adapta a este caso es almacenar las variables en una tabla unidimensional, empleando una constante para referirnos a una casilla en particular.

En un segundo paso habrás de plantearte cómo visualizarás estas figuras en la pantalla. Lo que equivale a determinar cuántos dibujos diferentes se van a necesitar a lo largo del juego. Todos los grandes juegos para ordenadores incorporan dibujos extremadamente cuidados y de gran colorido, lo cual quiere decir que este segundo paso no has de tomarlo a la ligera. Para bajar a un buen nivel de detalle, construiremos la figura de Freddy con una ordenación tres por dos de caracteres UDG. Para visualizar la araña utilizaremos figuras dos por dos (dispondremos de dos figuras para que la



«ABISMO»: AÑADIENDO NUEVAS INSTRUCCIONES

- TEMPLANDO LOS INSTRUMENTOS
- LA ENVOLVENTE
- DEFINICION DEL TONO
- DURACION DE LAS NOTAS
- «MANGAS VERDES»

Después de que ha aparecido en la pantalla la primera página con el título del programa y cuando el jugador sabe ya qué juego se supone que está jugando hay que decirle la forma en que lo tiene que hacer.

Nuevamente se utiliza un programa en BASIC para «pokear» letra a letra los datos de las instrucciones en una tabla ASCII cuando se ejecute el programa.

```
10 AD=17312:I=0
15 POKE AD+I,147:I=I+1
50 N=3:GOSUB 2000
60 POKE AD+I, 31:I=I+1
70 A$='LA HISTORIA
   CUENTA...':N=24:GOSUB
   1000
80 N=1:GOSUB 2000
90 A$='[ 18*COMM+T]'
   :N=18:GOSUB 1000
1000 N=3:GOSUB 2000
110 A$=' DESPUES DE UN
    CORTO PASEO, WILLY
    VUELVE ':N=40:GOSUB
    1000
120 A$='PARA BUSCAR LAS
    CABRAS QUE HAN
    ESPARCIDO':N=40:GOSUB
    1000
1250 A$='LA MERIENDA POR
     LAS ROCAS DEL
     ACANTILADO': N=40:
     GOSUB 1000
140 A$='WILLY QUIERE
    RECUPERAR SUS
    POSESIONES[ 3
    ESPACIOS]':N=40:GOSUB
    1000
150 A$='PERDIDAS, PERO
    DEBE LUCHAR CONTRA
    LOS[ 3 ESPACIOS]'
    :N=40:GOSUB 1000
160 A$='GUIJARROS, LAS
    GRIETAS Y LAS
    SERPIENTES
    ':N=40:GOSUB 1000
170 A$='VENENOSAS. POR SI
    FUERA POCO, LA MAREA[ 2
    ESPACIOS]':N=40:GOSUB
    1000
180 A$='ESTA SUBIENDO Y EL
    NO PUEDE NADAR.[ 6
    ESPACIOS]':N=40:GOSUB
    1000
190 A$='[ 3 ESPACIOS]PARA
    AYUDAR A WILLY EN ESTA
    AVENTURA ':N=40:GOSUB
    1000
200 A$='[ 3 ESPACIOS]LEE LAS
    SIGUIENTES
    INSTRUCCIONES Y[ 3
    ESPACIOS]':N=40:GOSUB
    1000
210 A$='PULSA S PARA JUGAR
    ':N=19:GOSUB 1000
220 N=3:GOSUB 2000
230 POKE AD+I,149:I=I+1
240 A$='[ 2
    ESPACIOS]TECLADO -
    SHIFT[ 3 ESPACIOS]=[ 3
    ESPACIOS]MOV. DERECHA[
    4 ESPACIOS]':N=40:GOSUB
    1000
250 A$='[ 16 ESPACIOS]z[ 3
    ESPACIOS]=[ 3
    ESPACIOS]SALTO
    VERTICAL. ':N=40:GOSUB
    1000
260 A$='[ 12
    ESPACIOS]AMBOS[ 3
    ESPACIOS]=[ 3
    ESPACIOS]SALTO A
    DERECHA.':N=40:GOSUB
    1000
270 N=1:GOSUB 2000
280 A$='[ 2
    ESPACIOS]JOYSTICK EN
    PORT 2[ 20
    ESPACIOS]':N=40:GOSUB
    1000
290 POKE AD+I,0
300 STOP
1000 REM *** RUTINA PARA EL
     TEXTO ***
1010 FOR J=1 TO N
1020 POKE AD+I,ASC(MID$(A$,
     J,1))
1030 I=I+1:NEXT :RETURN
2000 REM *RUTINA PARA
     INSERTAR RETURN*
2010 FOR J=1 TO N:POKE
     AD+I,13:I=I+1:NEXT
     :RETURN
```

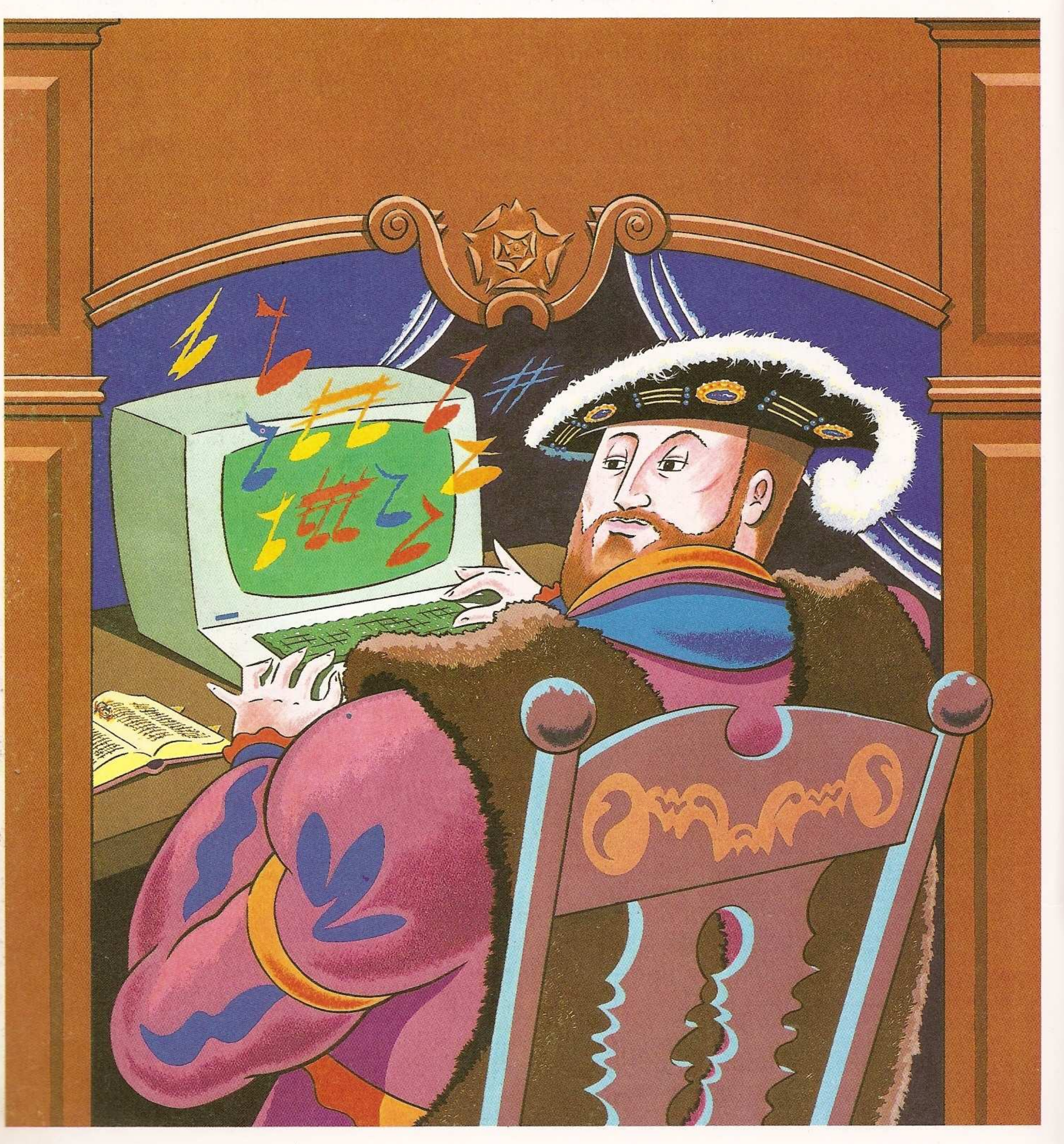
El siguiente programa se encarga de extraer los datos y presentarlos en la pantalla.

```
., 4060 A9 09      LDA #$09
., 4062 8D 20 D0   STA $ D020
., 4065 A9 0F      LDA #$0F
., 4067 8D 21 D0   STA $D021
., 406A A9 A0      LDA #$A0
., 406C 85 FB      STA $FB
., 406E A9 43      LDA #$43
., 4070 85 FC      STA $FC
., 4072 A0 00      LDY #$00
., 4074 EA         NOP
., 4075 B1 FB      LDA ($FB),Y
., 4077 F0 0E      BEQ $4087
., 4079 20 D2 FF   JSR $FFD2
., 407C EA         NOP
., 407D E6 FB      INC $FB
., 407F D0 02      BNE $4083
., 4081 E6 FC      INC $FC
., 4083 18         CLC
., 4084 90 EF      BCC $4075
., 4086 EA         NOP
., 4087 20 E4 FF   JSR $FFE4
., 408A C9 53      CMP #$53
., 408C D0 F9      BNE 4087
., 408E 60         RTS
```

Estos dos programas tienen que ser ensamblados y almacenados juntos, igual que hicimos con los programas de la primera parte de ABISMO. Si

no dispones de un ensamblador, os presentamos un programa cargador en BASIC para que podáis entrar los datos.

```
1Ø S=Ø
2Ø FOR AD=1648Ø TO 16526
3Ø READBY:POKEAD,
   BY:S=S+BY
4Ø NEXT
5Ø IFS<>7432 THEN
   PRINT'ERROR EN LOS
   DATOS':END
1648Ø DATA 169,9,141,32,2Ø8,
      169
16486 DATA 15,141,33,2Ø8,
      169.16Ø
16492 DATA 133,251,169,67,
      133,252
16498 DATA 16Ø,Ø,234,177,
      251,24Ø
```



```
165Ø4 DATA 14,32,21Ø,255,
      234,23Ø
1651Ø DATA 251,2Ø8,2,23Ø,
      252.24
16516 DATA 144,239,234,32,
      228,255
16522 DATA 2Ø1,83,2Ø8,249,
      96
```

Además de los dos métodos utilizados para suministrar los datos a las palabras impresas en la página de presentación del juego, el programa en BASIC que tenemos ahora utiliza una tercera manera de construir la tabla ASCII. En esta ocasión se considera a las palabras como cadenas de caracteres, pokeando el valor ASCII de cada letra en la tabla por medio de la subrutina que comienza en la línea 1000.

Pero antes que nada tienes que borrar la pantalla. Para ello, en la línea 15 se pokea en la tabla el valor 147, que es el valor ASCII correspondiente a la tecla ACLR/HOME. A continuación la línea 50 pone tres retornos de carro en la tabla, con fines de formateo y la línea 60 pokea en la tabla el código de la tinta azul.

La línea 290 pokea un 0 en la última posición de la tabla, de forma que el programa en código máquina sabe cuando llega al final.

EL PROGRAMA EN LENGUAJE ENSAMBLADOR

En la dirección D020 se almacena el número 0, con lo que al borde de pantalla se le asigna el color marrón. Seguidamente el color de pantalla, controlado por la posición de memoria D021, se pone a F que corresponde al gris.

La dirección de comienzo de los nuevos datos se almacena entonces en las direcciones FB y FC de la página cero y el registro de índice Y se pone a cero.

LDA ($FB), Y , realiza la carga de la tabla de datos a razón de un byte cada vez, utilizando direccionamiento indirecto. BEQ $4087 fuerza la salida de la rutina cuando se carga un cero en el acumulador, en otras palabras, cuando se llega al final de la tabla el procesador pasa a la siguiente parte del programa.

También aquí se utiliza la subrutina de ROM que hay en FED2 para que aparezca en la pantalla el carácter cuyo código ASCII contiene el acumulador.

La pequeña rutina que viene a continuación incrementa el puntero en FB y FC. La instrucción BNE $4083 fuerza un salto sobre la instrucción que incrementa el byte alto del puntero. El puntero queda incrementado, mientras que el registro de índice Y se mantiene a cero, debido a que la tabla de datos tiene una longitud superior a los 255 bytes.

El indicador de acarreo queda entonces puesto a cero y el procesador hace un salto hacia atrás para cargar el siguiente byte de la tabla.

ESPERANDO EL FINAL

La rutina de ROM que hay en FFE4 es la rutina GETIN que toma un carácter de la cola en el buffer de teclado. El valor ASCII del carácter se pasa al acumulador y la instrucción CMP #53 compara su valor con el valor hexadecimal 53; que es el código ASCII de la letra 8.



Si no se ha pulsado una S y no figura este carácter ASCII en el buffer del teclado, la instrucción BNE $4087 envía al procesador a este pequeño bucle a fin de comprobar el siguiente byte del buffer de teclado.

Pero si, por el contrario, se pulsó una S, no se cumple la condición BNE y el procesador pasa a la siguiente rutina (sólo en este caso se produce la rotura forzada por la instrucción BRK y se detiene debido al RTS).

TEMPLANDO LOS INSTRUMENTOS

Comenzaremos por preparar una melodía antes de que Willy se apresure a partir hacia la aventura. Sí, en INPUT COMMODORE hemos vuelto a la década de los 60, queremos decir a la de 1560, con una tonadilla que nos servirá de motivo musical para nuestro juego del ABISMO.

Ningún juego puede considerarse completo mientras no se le incorpore alguna musiquilla agradable. En el juego del ABISMO hemos introducido la música de *Mangas Verdes* ("Greensleeves") melodía atribuida al rey Enrique VIII de Inglaterra, que se repetirá cada cierto tiempo.

La rutina que interpreta la melodía la hemos puesto aparte de la rutina principal, debido a que será llamada varias veces desde varias partes diferentes del programa. Pero mientras esperas después de que haya desaparecido la página de presentación, se inserta la siguiente cuña:

```
10  S=0
20  FOR AD=16473 TO 16478
30  READBY:POKEAD,
    BY:S=S+BY
40  NEXT
50  IFS<>668 THEN
    PRINT'ERROR EN LOS
    DATOS':END
16473  DATA 169,19,32,150,64,
       234
```

Y en lenguaje ensamblador:

```
4059  A9 13       LDA #$13
405B  20 96 40    JSR $ 4096
405E  EA          NOP
```

Esto carga el acumulador con el número 13 hexadecimal, que es 19 en decimal, ya que hay 19 notas en la frase inicial de *Mangas Verdes*. A continuación salta a la subrutina que ejecuta la melodía, que está en 16534.

```
4096 C9 13        CMP #$13
4098 90 03        BCC $409D
409A A9 13        LDA #$13
409C EA           NOP
409D EA           NOP
409E EA           NOP
409F EA           NOP
40A0 85 FB        STA $FB
40A2 A0 18        LDY #$18
40A4 A9 00        LDA #$00
40A6 99 00 D4     STA $D400,Y
40A9 88           DEY
40AA D0 FA        BNE $40A6
40AC EA           NOP
40AD A9 0F        LDA #$0F
40AF 8D 18 D4     STA $D418
40B2 EA           NOP
40B3 A9 00        LDA #$00
40B5 8D 05 D4     STA $D405
40B8 A9 F0        LDA #$F0
40BA 8D 06 D4     STA $D406
40BD A0 00        LDY #$00
40BF B9 60 43     LDA $43G0,Y
40C2 8D 01 D4     STA $D401
40C5 C8           INY
40C6 B9 60 43     LDA $4360,Y
40C9 8D 00 D4     STA $D400
40CC C8           INY
40CD 84 FC        STY $FC
```



```
4ØCF A9 ØØ       LDA #$ØØ
4ØD1 AA          TAX
4ØD2 A8          TAY
4Ø D3 2Ø DB FF   JSR $FFDB
4ØD6 A4 FC       LDY $FC
4ØD8 EA          NOP
4ØD9 B9 6Ø 43    LDA $436Ø,Y
4ØDC C8          INY
4ØDD 84 FC       STY $FC
4ØDF 85 FD       STA $FD
4ØE1 EA          NOP
4ØE2 A9 11       LDA #$11
4ØE4 8D Ø4 D4    STA $D4Ø4
4ØE7 EA          NOP
4ØE8 2Ø DE FF    JSR $FFDE
4ØEB C5 FD       CMP $FD
4ØED DØ F9       BNE $4ØE8
4ØEF EA          NOP
4ØFØ A4 FC       LDY $FC
4ØF2 A9 ØØ       LDA #$ØØ
4ØF4 8D Ø4 D4    STA $D4Ø4
4ØF7 EA          NOP
4ØF8 C6 DB       DEC $FB
4ØFA DØ C3       BNE $4ØBF
4ØFC 8D 18 D4    STA $D418
4ØFF 6Ø          RTS
```

El cargador en BASIC sería:

```
1Ø S=Ø
2Ø FOR AD=16534 TO 16639
3Ø READBY:POKEAD,
   BY:S=S+BY
4Ø NEXT
5Ø IFS<>16ØØ1 THEN PRINT'
   ERROR EN LOS DATOS':END
16534 DATA 2Ø1,19,144,3,169,
      19
1654Ø DATA 234,234,234,234,
      133,251
16546 DATA 16Ø,24,169,Ø,153,
      Ø
16552 DATA 212,136,2Ø8,25Ø,
      234,169
16558 DATA 15,141,24,212,
      234,169
16564 DATA Ø,141,5,212,169,
      24Ø
1657Ø DATA 141,6,212,16Ø,Ø,
      185
16576 DATA 96,67,141,1,212,
      2ØØ
16582 DATA 185,96,67,
      141,Ø,212
16588 DATA 2ØØ,132,252,169,
      Ø,17Ø
16594 DATA 168,32,219,255,
      164,252
166ØØ DATA 234,185,96,67,
      2ØØ,132
166Ø6 DATA 252,133,253,234,
      169,17
16612 DATA 141,4,212,234,32,
      222
16618 DATA 255,197,253,2Ø8,
      249,234
16624 DATA 164,252,169,Ø,
      141,4
1663Ø DATA 212,234,198,251,
      2Ø8,195
16636 DATA 141,24,212,96
```

LIBRE DE ERRORES

El primer pequeño módulo de este programa está prote-

gido contra una entrada errónea en los datos. Cuando se llama a esta rutina desde alguna otra parte del programa principal, puede ser que se te olvide cargar el acumulador adecuadamente. Como sólo hay datos suficientes para asegurarse de que la rutina no está siendo llamada para interpretar más notas. Por eso se compara el contenido del acumulador con 19.

Pero si el contenido del acumulador es superior a 19, no se produce tal salto, y el acumulador se carga con el valor 19. Esto significa que se puede llamar a la rutina para tocar menos de 19 notas, pero no más. El número de notas que hay que interpretar se almacena entonces en FB, de modo que se puede utilizar el acumulador para inicializar el chip SID.

EL CHIP «SID»

El circuito integrado SID 6581 de Commodore, del que ya nos hemos ocupado en INPUT, controla la síntesis del sonido y de la música. Las direcciones 54272 a 54296 corresponden a dicho chip y cuando vayas a utilizarlo, lo primero que tienes que hacer es borrarlas poniendo ceros en sus 24 direcciones.

Por ello se carga el valor 24 (18 en hexadecimal) en el registro de índice Y, cargándose además un 0 en el acumulador. Después se carga un cero en 54272 (D400 en hexadecimal), con un desplazamiento equivalente al contenido de Y, lo que da el valor 54296. A continuación se decrementa Y regresando el ordenador hacia atrás.

La última dirección del SID, la 54296 (D418) controla el volumen global de la salida del sonido. El máximo volumen se obtiene cargando esta última dirección del SID 6851 con el valor 15.

LA ENVOLVENTE

Se llama envolvente a la forma en que varía el volumen de un sonido durante el tiempo que dura. Ya hemos hablado en otros números de INPUT de la manera de configurar la envolvente de un sonido. Por ahora, únicamente necesitas recordar que hay cuatro parámetros que intervienen:



ataque, caída, sostenimiento y soltado.

El ataque de una nota caracteriza la velocidad a la que alcanza su máximo volumen. El volumen medio es lo que llamamos su sostenimiento. La velocidad a la que disminuye su volumen a partir del volumen medio es lo que hemos llamado «soltar».

Puedes entender fácilmente el significado de todos estos términos si piensas en la forma en que se ejecuta una nota tocada en un piano. El ataque está controlado por la fuerza con que aprietes la tecla. Si le das bruscamente, el ataque es brusco; si la golpeas con más suavidad, el ataque será más dulce.

Después del golpe inicial del macillo contra la cuerda, el volumen disminuye con bastante rapidez. Pero mientras mantengas la tecla bajada los apagadores estarán separados de la cuerda y la nota se mantendrá. El sonido se extingue completamente en cuanto el apagador reposa de nuevo sobre la cuerda al soltar la tecla.

Los cuatro atributos de ataque, caída, sostenimiento y soltado pueden adoptar valores cuantificados entre 0 y 15. El valor del ataque se almacena en los cuatro bits de mayor peso de la dirección de memoria D405 y el valor de caída en los cuatro bits de menor peso de esa misma dirección. Los valores de sostenimiento y soltado se almacenan respectivamente en los niveles alto y bajo de la dirección de memoria D406.

Los valores de ataque, caída y soltado se ponen a cero, mientras que el valor de sostenimiento se fija en 15.

DEFINIENDO EL TONO

El tono de los sonidos está controlado por las direcciones de memoria D400 y D401. Los detalles para el tono de cada nota de la frase inicial de *Mangas Verdes* están contenidos en la tabla de datos que hay en 4360 hexadecimal.

El registro de índice Y se carga con el valor cero y el primer byte de la tabla de datos se carga sobre el byte alto de la dirección de control de tono. A continuación se incrementa Y, cargándose el siguiente byte de datos en el byte bajo de la posición de control de tono; después se incrementa nuevamente Y. El índice se almacena entonces en FC.

LA DURACION DE LAS NOTAS

También tienes que especificar lo que dura cada nota. Pero no hay ninguna dirección en el circuito integrado SID que controle dicha duración. Tienes que ocuparte tú de ello, extinguiendo la nota cuando quieras que cese el sonido.

Para ello, lo primero que tienes que hacer es poner a cero el reloj de tiempo real. Este reloj está controlado por la rutina SETTIM en la ROM Kernal. Se carga un cero en el acumulador y se transfiere a los registros X e Y. Estos tres registros tienen que estar a cero para inicializar el reloj. El procesador salta entonces a SETTIM, que empieza en FFDB. La duración que debe tener la nota es el tercer parámetro de la tabla de datos. Por ello se restaura el índice a partir de FC y a continuación se carga en el acumulador el siguiente byte de datos. Seguidamente se incrementa el registro de índice Y, almacenándose en FC, quedando almacenada la duración de la nota en FD.

La dirección de memoria D404 es el registro de control del circuito SID y cada uno de sus bits controla una función determinada. En este caso se ha cargado con un 17 (11 en hexadecimal), que en binario es 00010001, con lo cual están activados los bits cero y cuatro.

El bit cero es la «puerta» (gate), de modo que cuando es activado empieza a emitir el sonido. El bit cuatro permite obtener un sonido con una forma triangular, que resulta con una calidad suave y aflautada.

Una vez que ha sido establecido el sonido, el procesador salta a la subrutina RDTIM, que está en la ROM Kernal, la cual realiza una lectura en el reloj de tiempo real. El reloj cuenta 1/60 de segundo y almacena el tiempo en tres bytes. Pese a lo que dice la *Guía de Referencia para Programadores*, cuando se lee el tiempo las cifras menos significativas son devueltas en el acumulador, las siguientes en el registro X y las más significativas en el registro Y.

Las notas musicales duran un tiempo muy corto, por lo que es el byte del acumulador el que se compara con el parámetro de duración que hay en FD. Si ambos valores no son iguales, o, lo que es lo mismo, si el reloj todavía no ha llegado al valor designado, la instrucción BNE fuerza un salto a la instrucción JSR, que impone una salida y una nueva lectura del tiempo. Pero cuando se alcanza el tiempo requerido, no se cumple la condición de la instrucción BNE y el procesador sigue avanzando.

De nuevo se restaura el registro de índice Y desde FC. Se carga un cero en el acumulador y se almacena en la dirección D404, que es el registro de control del SID. Así se apaga nuevamente el sonido.

A continuación se decrementa el contador de notas que hay en FB y si todavía no ha llegado a cero, la instrucción BNE recorre el bucle para ocuparse de la siguiente nota. Si la cuenta ya ha llegado a cero, el contenido del acumulador, que todavía es cero, se almacena en D418, con lo cual el volumen se pone a cero. En este momento el procesador regresa al BASIC, ya que acabamos de llegar al final de la primera parte de Abismo.

LOS DATOS

La anterior rutina de lenguaje ensamblador puede utilizarse para ejecutar cualquier melodía, siempre que modifiques el valor del contador de notas que hay en FB. Puedes cargar la forma de la envolvente y el tipo de sonido producido, realizando los oportunos ajustes de los valores depositados en las direcciones D404, D405 y D406.

Pero no se interpretará absolutamente nada a menos que suministres algunos datos. Tampoco aquí vale la pena hacerlo en código máquina; el siguiente programa en BASIC pokea los datos de *Mangas Verdes* en una tabla de datos a la que pueda tener acceso el *programa en código máquina*.

```
10 ADD=17248:FORI=0TO32000
   20 READA%:POKEADD+I,A%
30 IFA%=0GOTO50
40 NEXT
50 END
17248 DATA 28,126,16,45,198,
      32,51,97,16,57
17258 DATA 172,64,61,126,16,
      57,172,32,51,97
17268 DATA 32,43,52,16,34,
      75,64,38,126,16
17278 DATA 45,52,32,45,198,
      32,38,126,16,38
17288 DATA 126,64,34,75,16,
      38,126,32,43,52
17298 DATA 32,36,85,16,28,
      214,32,0
```

LISTA DE PREMIADOS EN LA ENCUESTA DE JUNIO

Primer premio de 25.000 pts. en metálico:

MANUEL VAZQUEZ RODRIGUEZ. Calahorra (La Rioja)

Diez premios de un juego para cada concursante:

MIGUEL ANGEL VAZQUEZ FERNANDEZ. Vigo
JUAN CARLOS SANTOS MARTIN. Guadalajara
DAVID JOU SALA. Barcelona
ADELARDO MANUEL DORADO CARO. Alcalá de H. (Madrid)
FERNANDO FERNANDEZ ABRIL. Ponferrada (León)
LUIS CHICHARRO HERNANZ. Madrid
MANUEL CARLOS PERALTA CAMPO. Lérida
JORDI JOAQUIN ESTEVE. Tarragona
RUBEN GONZALEZ LILLO. Navarra
MANUEL RACHAD HAJ-SALEH RAMIREZ. Sevilla

CUADROS CON GRAFICOS DEFINIDOS POR EL USUARIO (I)

Una vez has definido un juego de figuras, es muy fácil crear una gran variedad de cuadros con sólo variar la disposición de los UDG y cambiar el fondo.

En el último artículo de la serie se indicaba la manera de que pudieses superar cualquier límite que tuviera tu ordenador en el número de UDG que pudieras crear, y cómo puedes definir el juego de caracteres. Este artículo es el primero de una serie de dos que te mostrarán cómo puedes ahorrar una gran cantidad de tiempo y de esfuerzo utilizando UDG para formar un cuadro en la pantalla. UDG son las siglas de *User Defined Graphics*, que significa «gráficos definidos por el usuario».

¿POR QUE HAY QUE EMPLEAR LOS UDG?

Supón que deseas crear un cuadro de la jungla —quizá una página de título para un nuevo y emocionante juego—. Para ello existen básicamente dos maneras de hacerlo. Podrías dibujar con DRAW cada parte del cuadro y colorearlo como deseases. Pero supón que quisieses un bosque como parte de tu fondo. Tendrías que dibujar un tronco y tres copas para cada árbol que incluyeses en el cuadro, o sea, una tarea realmente tediosa. Y, desde luego, cada uno de estos troncos y copas serían iguales. En cambio, si puedes diseñar uno o varios UDG, que se combinen para formar un árbol, sólo tendrás que colocar con un PRINT las figuras recién creadas.

Este último procedimiento tiene la evidente ventaja de que te ahorra la necesidad de especificar cada uno de los detalles cada vez que deseas tener un árbol en la pantalla, además de otras ventajas adicionales.

Una de ellas es que ahorras tiempo. Los ordenadores presentan caracteres con PRINT mucho más rápidamente que con DRAW trazando líneas de alta resolución (lo cual también es verdad si los caracteres son definidos por el usuario y muy complicados). Por tanto, si almacenas tu cuadro en forma de UDG, cuando el ordenador llega a la parte de tu programa que presenta el cuadro, no tienes que estar aguardando un largo tiempo mientras el ordenador lleva trabajosamente tu obra a la pantalla: aparece casi instantáneamente.

LIBERTAD DE ELECCION

Otra ventaja es que puedes variar fácilmente, y cuando lo desees, el tamaño del objeto. Con el ejemplo anterior puede variar la altura del árbol simplemente añadiendo o quitando uno o más UDG del «tronco», o alterar su follaje empleando una combinación distinta de UDG de «hoja». Naturalmente, esto también puedes hacerlo con los comandos de alta resolución de tu ordenador, pero los UDG son mucho más cómodos. Para empezar, tratas con los números mucho más manejables de la pantalla de baja resolución.

Una vez has definido los UDG para tu cuadro, permanecen en la memoria (a menos que los cambies), aunque no son necesaria ni permanentemente accesibles desde el teclado; si tienes varios bancos o juegos de caracteres al mismo tiempo en la memoria, tendrás que «activar» el banco que debe ser accesible desde el teclado. Para esto bastará simplemente con variar el puntero de los UDG.

Esto significa que podrás utilizar las figuras que has diseñado tantas veces como desees en tus programas: el único límite es la memoria del ordenador. En consecuencia, en tu pro-



- COMO DECIDIR CUANDO HAY QUE UTILIZAR LOS UDG
- FORMACION DE UN CUADRO CON VARIOS CARACTERES
- REALIZACION DE UN CUADRO DE LA JUNGLA
- UN COCODRILO, UN ELEFANTE Y ARBOLES
- DIBUJO DEL FONDO
- ALTA Y BAJA RESOLUCION

grama resultará tan fácil tener un cuadro de bosque como un solo árbol.

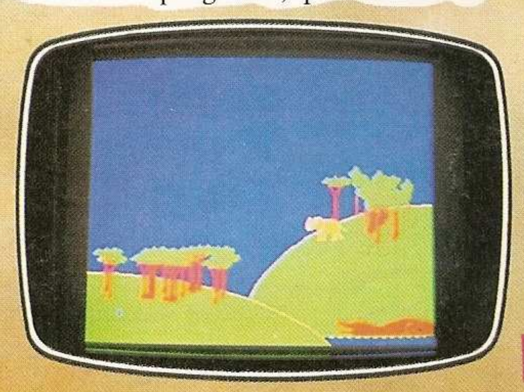
No obstante, el empleo de los UDG tiene algunos pequeños inconvenientes. Para empezar, tienes que definir todas las figuras y teclear los datos en el ordenador, aunque, normalmente, esto no es peor que trabajar con gráficos de alta resolución. Pero también hay que utilizar una parte de la limitada memoria de tu ordenador para almacenar los UDG y, a menos que seas muy cuidadoso, acabarás por almacenarlos dos veces. En el segundo artículo de esta serie se indica cómo puedes evitar esta repetición de memoria.

¿QUE IMAGENES EMPLEAN LOS UDG?

Por las razones expuestas anteriormente, hay una serie de imágenes que puedes dibujar de manera mucho más eficiente con UDG que otras.

Como regla general práctica, si el cuadro tiene un número relativamente pequeño de objetos macizos en partes que aparecen con una forma similar varias veces, o si deseas emplear una imagen varias veces durante la ejecución de un programa, podrás ahorrar tiempo y esfuerzo utilizando los UDG.

Pero si quieres tener un cuadro muy detallado que sólo vas a emplear una vez en el programa, probablemente



será mejor que utilices los comandos de alta resolución del ordenador.

Por ejemplo, si el cuadro debe tener como telón de fondo una pared de ladrillos, la podrás formar rápidamente empleando sencillamente uno o dos UDG que presentarás con PRINT repetidamente en la parte del cuadro en la que desees que aparezca la pared. Una alternativa podría ser dibujar una serie de líneas en una zona coloreada para simular las juntas de los ladrillos.

Otro ejemplo sería el cuadro de la jungla mencionado anteriormente. Ya se ha indicado que puedes ahorrar tiempo definiendo los UDG para los árboles que debe contener cualquier escenario de jungla. Los animales que deben acompañar a los árboles también son muy adecuados para ser definidos con UDG, puesto que así podrás emplearlos en cualquier punto del programa, para animarlos, o para disponer de un gran número de los mismos.

UN CUADRO DE LA JUNGLA

A continuación se presenta un programa que sirve para definir los caracteres típicos que probablemente desearás incluir en la escena de la jungla.

```
1Ø POKE 52,48:POKE 56,
   48:CLR
2Ø POKE 56334,Ø:POKE
   1,35
3Ø FOR Z=Ø TO 1Ø24:POKE
   12288+Z,PEEK(53248+Z):
   NEXT Z
4Ø POKE 1,39:POKE 56334,
   1
5Ø FOR Z=Ø TO 143:READ
   X:POKE 13312+Z,X:
   NEXT Z
52 FOR Z=76Ø TO 983:READ
   X:POKE 13312+Z,X:
   NEXT Z
55 C$=CHR$(13):POKE 53272,
   28:POKE 53281,Ø:POKE
   5328Ø,6
6Ø PRINT
   "[SHIFT+CLR/HOME]":FOR
   Z=1 TO 3Ø:X=INT
   (RND(1)*32Ø)
```





```
70 POKE 1024+X,46:POKE
   55296+X,RND(1)*6+2:NEXT
   Z
80 T$="[CTRL+9][CTRL+6][C
   OMM+D][COMM+Z][CRS
   R ABAJO][ 2*CRSR IZQDA.
   ][COMM+S][COMM+@][CR
   SR ABAJO][ 2*CRSR IZQD
   A.][COMM+A][COMM+E][C
   RSR ABAJO][CRSR IZQDA.]
   [CTRL+5][COMM+R]":C(0)
   =3
85 TT$="[CTRL+4][CTRL+9][
   COMM+W][COMM+G][COM
   M+H][COMM+L][CTRL+3][
   CRSR ABAJO][ 3*CRSR IZ
   QDA.][COMM+Y][COMM+U]
   [CRSR ABAJO][ 2*CRSR IZ
   QDA.][COMM+O][CRSR AB
   AJO][CRSR IZQDA.][SHIFT
   +@]":C(1)=6
87 PRINT "[CLR/HOME][
   7*CRSR
   ABAJO][CTRL+6]":FOR Z=1
   TO 280:PRINT "[COMM+T]";
   :NEXT
90 PRINT "[CLR/HOME][ 6*CR
   SR ABAJO]"TAB(13)TT$"
   [ 4*CRSR ARRIBA][ 2*CRS
   R DCHA.]"T$"[ 2*CRSR A
   RRIBA][CRSR DCHA.]"T$TT
   $
95 PRINT "[CLR/HOME][ 4*CR
   SR ABAJO]"TAB(30)T$"[ 2
   *CRSR ARRIBA]"T$"[ 2*C
   RSR ARRIBA]"
   T$TT$
96 PRINT "[CLR/HOME][ 9*CR
   SR ABAJO][CTRL+5]"SPC(
   30)"[CTRL+9][COMM+ *]"
   CHR$(160)"[COMM+K][CO
   MM+I][CRSR ABAJO][ 4*C
   RSR IZQDA.][CTRL+9][CO
   MM+T][COMM+@][COMM+
   G][COMM+ +][COMM+M][
   CRSR ABAJO][ 5*CRSR IZ
   QDA.][CTRL+9][CRSR DCH
   A.][COMM+£][SHIFT+£][C
   OMM+N][COMM+
   Q]"
150 FOR Z=0 TO 159:POKE
    1864+Z,102:POKE
    56136+Z,C(RND(1)*2):
    NEXT Z
160 PRINT "[CLR/HOME][
    20*CRSR ABAJO]"TAB
    (28);
162 IF RND(1)>.50 THENPRI
    NT "[CTRL+9][CTRL+8]@
    ABCDEFGHIJK";SPC(28);"
    [CTRL+9]LMNOOOOOPOO
    O[CTRL+0]";:GOTO
    170
164 PRINT "[CTRL+8][ 12 ES
    PACIOS]";SPC(28);"[CTRL
    +9]@ABCDEFGHIJK[CTRL
    +0]";
170 FOR D=1 TO 200:NEXT
    D
190 GOTO 160
```

Si introduces y haces correr el programa en el ordenador, podrás ver la clase de cuadro que podrás crear con UDG. No te preocupes si la mitad superior de la pantalla aparece algo vacía en este momento, ya que en el próximo artículo de esta serie se terminará el cuadro.

VERSATILIDAD DE LOS UDG

Podrás comprobar la versatilidad de los UDG observando el agua que hay debajo del cocodrilo. En realidad es un solo UDG que se repite en toda la zona del agua.

También podrás observar que el cuadro no sólo contiene UDG, sino también algunas líneas de alta resolución que dan la impresión de que en el cuadro también hay colinas.

En este ejemplo puedes comprobar que si bien los UDG tienen muchas ventajas con respecto a los comandos de alta resolución, a menudo podrás obtener muy buenos resultados empleando ambos procedimientos en el mismo cuadro.

En el siguiente artículo de esta serie se añadirán más cosas al cuadro. Por tanto, salva esta parte en la cinta para evitar tener que escribirlo otra vez.

COMO FUNCIONA EL PROGRAMA

El programa se ha dividido en dos partes. En la primera parte se definen

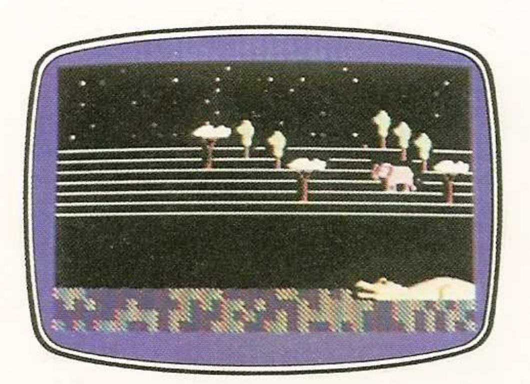
todos los caracteres y, en la segunda, los UDG se presentan en las posiciones adecuadas.

El programa empieza con varios POKE para leer el juego de caracteres de la ROM e introducirlo en un área protegida de la RAM.

Después continúa introduciendo los datos de cada uno de los caracteres definidos por el usuario en la RAM, con lo que ya puedes utilizar los caracteres. Esta operación termina en la línea 52.

La línea 55 establece que la variable C$ sea igual a CHR$(13), que es un RETURN. Esto se hace para mover la posición de PRINT a la siguiente línea de la pantalla durante la presentación efectiva del cuadro. La línea también establece el modo y los colores del fondo y del marco con tres POKE.

La pantalla se borra en la línea 60, y el ordenador prosigue iniciando la presentación del cuadro. En primer lugar presenta aleatoriamente una serie de estrellas, puesto que el cuadro representado es «La jungla de noche». Estas estrellas son simplemente puntos presentados en colores elegidos aleatoriamente mediante el bucle FOR...NEXT de las líneas 60 y 70.



Seguidamente, el programa presenta los UDG en las posiciones determinadas por los caracteres de posicionamiento de cursor que pueden verse en cada sentencia PRINT.

Los UDG de los árboles son los que se presentan en primer lugar y, para mayor facilidad, los caracteres que los forman se asignan a dos variables de cadena, T$ y TT$. Entonces, las líneas 90 y 95 imprimen las dos cadenas.

La línea 87 añade algo más de fondo, que son las líneas horizontales que puedes ver al ejecutar el programa. El otro animal estático, el elefante, lo presenta la línea 96.

El cocodrilo se presenta mediante una rutina algo más compleja para que se mueva hacia arriba y hacia abajo en el agua. Esta rutina ocupa las líneas

160 y 190, y simplemente imprime una línea de espacios encima de la parte inferior del cocodrilo si el número elegido aleatoriamente por la línea 162 es mayor que 0.5.

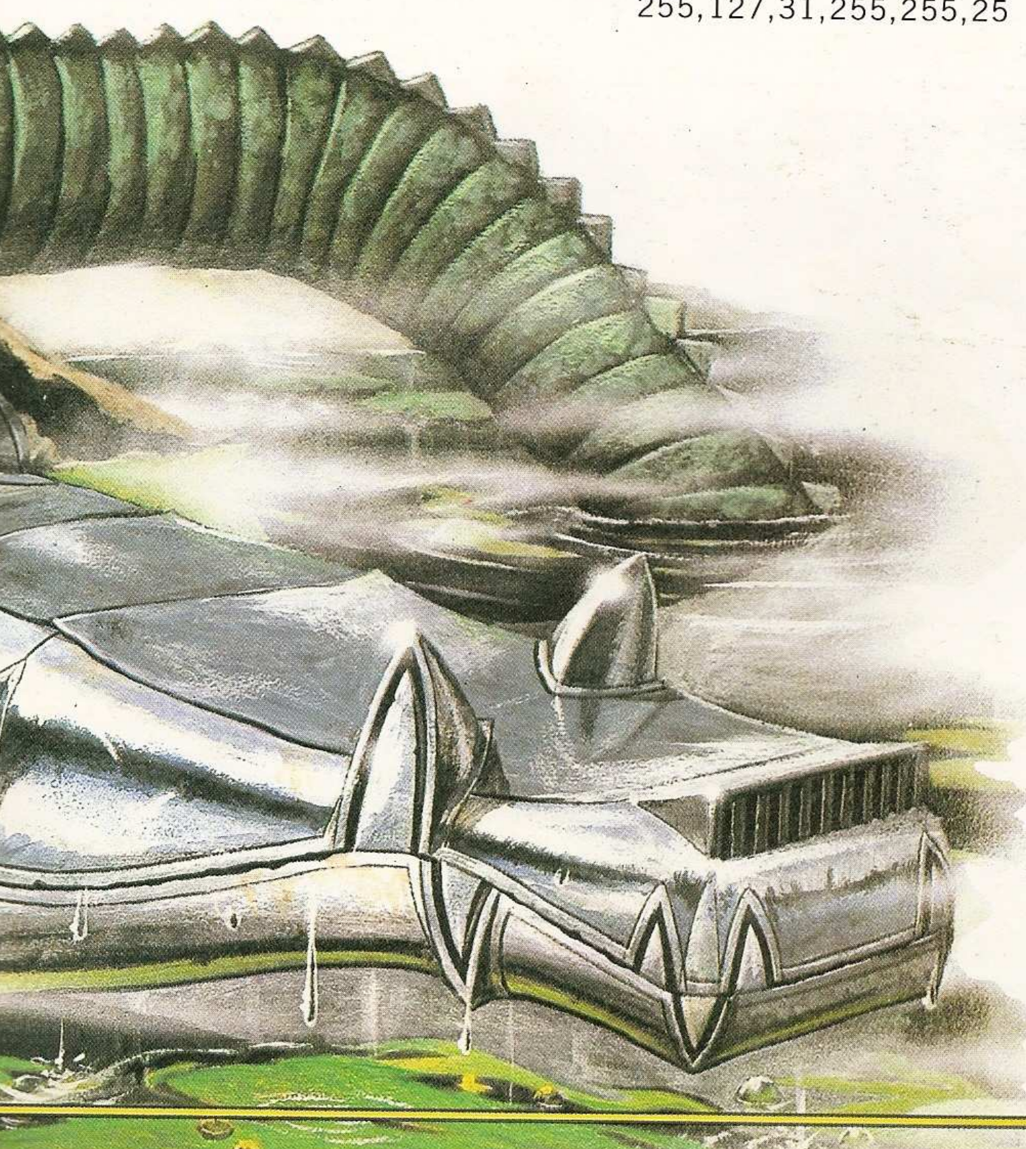
Si no te gusta que el cocodrilo se mueva, puedes dejarlo quieto cambiando la línea 190 a:

190 GOTO 190

UNA MANADA DE ELEFANTES

Es posible que prefieras sustituir el elefante único que se presenta en este momento por toda una manada. Esto podrás hacerlo de manera parecida al método empleado para presentar varios árboles: con un bucle FOR...NEXT o con varios GOSUB. En la siguiente parte de este artículo se verá cómo puedes añadir más figuras a la escena, así como aplicar alguna animación a la misma.

```
1300 REM COCODRILO
1310 DATA 0,0,1,7,15,15,9,5,0
     ,0,128,192,248,255,12
     7,95,1,3,6,12
1320 DATA 62,255,255,255,
     192,224,176,159,191,2
     55,255,255
1330 DATA 0,0,0,0,0,248,252,
     255,0,0,0,0,0,1,207,0
     ,0,0,1,15,127,255,255
1340 DATA 0,3,63,255,255,
     255,255,255,127,255,2
     55,255,255,254,249,24
     7,248
1350 DATA 255,255,255,255,
     15,255,255
1360 DATA 0,224,254,255,255,
     255,255,255,0,0,0,192,
     248,255,255,255,0,2,4,
     7,7
1370 DATA 3,0,0,21,1,164,73,
     255,255,0,0
1380 DATA 255,127,63,63,255,
     255,127,31,255,255,25
     5,255,255,255,255,255
1390 DATA 239,239,239,239,
     239,247,247,247,60,25
     5,255,255,255,255,255
     ,255
1780 REM ELEFANTE
1790 DATA 0,0,0,8,28,25,51,
     51,0,0,0,126,255,193,2
     53,253,0,0,0,0,0,255,2
     55
1800 DATA 255,0,0,0,0,0,248,
     252,254
1810 DATA 102,111,111,111,1
     25,57,26,0,253,251,25
     1,251,231,31,15,15,25
     5,255
1820 DATA 255,255,255,255,
     255,255,254,255,255,2
     55,255,255,255,255
1830 DATA 0,0,128,64,32,16,1
     2,0,15,15,15,14,14,14,
     14,31,255,240,224,224
1840 DATA 224,224,224,224,2
     55,63,59,59,57,57,121,
     248
1850 DATA 0,0,128,192,224,
     224,128,0
1860 REM ARBOL 1
1870 DATA 0,0,0,0,1,1,3,7,0,0
     ,0,0,224,240,248,248,1
     5,63,63,63,31,15,3,3
1880 DATA 252,254,254,254,
     254,254,252,252
1890 DATA 3,3,3,3,3,1,0,0,248
     ,248,248,248,248,248,
     240,96,96,96,96,96,96,
     96
1900 DATA 96,96
1910 REM ARBOL 2
1920 DATA 0,3,15,31,127,127,
     63,1,7,15,255,255,255,
     255,255,255,15,63,255
     ,255
1930 DATA 255,255,255,255,0
     ,128,248,248,248,248,
     240,224
1940 DATA 255,227,96,48,24,
     25,13,15,252,240,96,9
     6,192,192,128,128,7,7,
     7,7,7
1950 DATA 7,7,7,7,7,7,15,15,
     15,31,63
```



BUSQUEDA DE ANOMALIAS

Deberás tener mucho cuidado al introducir los datos, puesto que es inevitable que haya una gran cantidad.

—Si tu ordenador se detiene al hacer correr el programa con un mensaje de error OUT OF DATA, significa que en el programa no hay suficientes datos.

—Si has indicado al ordenador que lea la cantidad correcta de datos, puede haber dos posibles razones para ello: puedes haber olvidado uno o más números o puedes haber introducido puntos en lugar de comas. Un punto en lugar de una coma cambia dos números en uno, puesto que el punto se interpreta como un punto decimal.

—La única solución para estos dos problemas es comprobar, y volver a comprobar, los datos hasta que encuentres el error. Resulta de mucha ayuda incluir un comando PRINT en el bucle que introduce los datos, con el que el ordenador presenta cada número al introducirlo. De esta manera podrás comprobar cada número a medida que se introduce.

—Si has introducido demasiados datos, o la cantidad correcta pero erróneos, el programa correrá sin ninguna dificultad, pero verás una serie muy extraña de imágenes: por ejemplo, el cocodrilo podrá tener una trompa o el elefante podrá parecerse a una copa de árbol. Como antes, la única solución consiste en comprobar todos los datos introducidos.

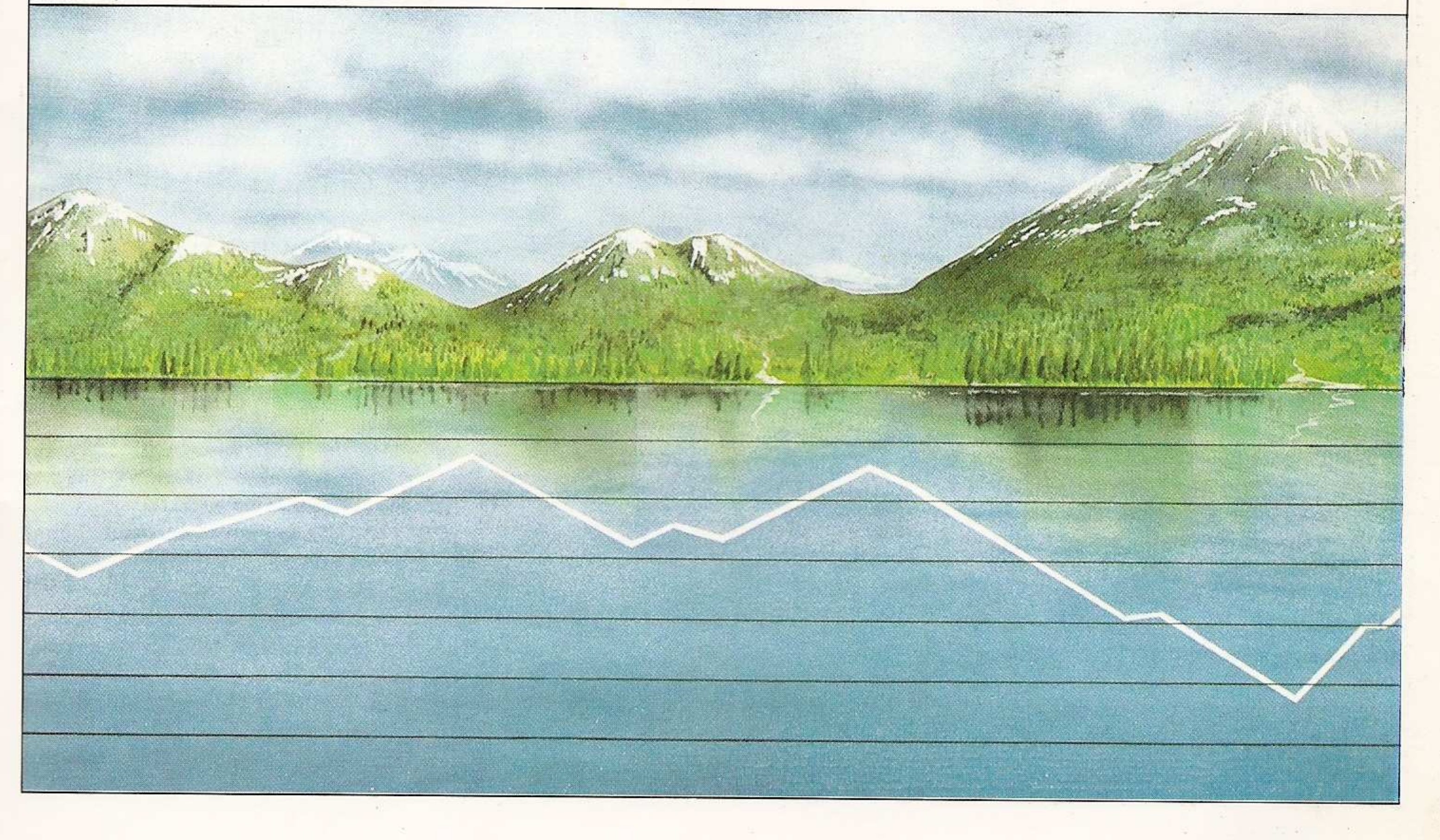
NO OLVIDES EL TELEFONO... ☎

Cuando, por cualquier motivo, nos escribas, no olvides indicar tu número de teléfono. Así nos será más fácil y rápido ponernos en contacto contigo. Gracias.

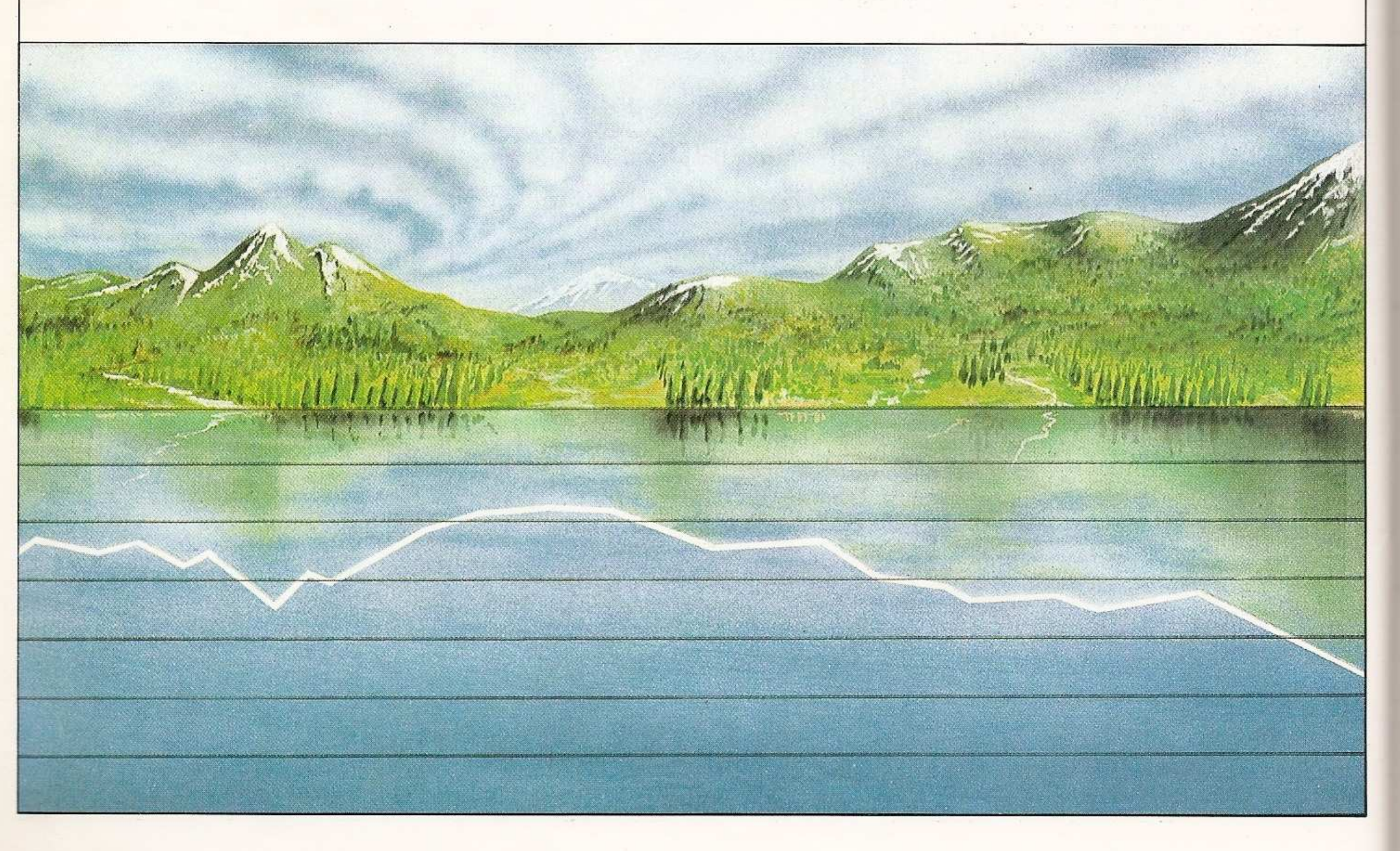
PROGRAMA MULTIGESTION

A continuación publicamos la segunda parte del Programa Multigestión que, por razones de espacio, no se publicó completo en el número anterior de INPUT.

```
1Ø35 PRINT MID$(V$,A,1);"[CRSR ABAJO]";
1Ø4Ø NEXT A
1Ø45 PRINT "[CLR/HOME] [ 23*CRSR ABAJO]";H$
1Ø5Ø FOR A=1 TO 15
1Ø55 X=A*2
1Ø6Ø IF D(A)=)Ø THEN1Ø8Ø
1Ø65 FOR Y=Ø TO D(A)
1Ø7Ø POKE 19Ø6+X-4Ø*Y,16Ø:POKE 56177+X-4Ø*Y,A
1Ø75 POKE 19Ø6+X-4Ø*Y,16Ø:POKE 56177+X-4Ø*Y,A:NEXT Y
1Ø8Ø NEXT A
1Ø85 GET KK$: IF KK$="  " THEN1Ø85
1Ø9Ø IF KK$="S" THEN55Ø
1Ø95 IF KK$="O" THEN95Ø
11ØØ IF NOT KK$="F" OR NOT KK$="O" THEN1Ø85
11Ø5 REM ***** ? HORA ? ******
111Ø TI$="ØØØØØØ":FOR J=1 TO 1ØØØ
1115 INPUT "[ 4 ESPACIOS] [SHIFT+H]ORA [ 2 ESPACIOS] (ØØØØØØ) =>";TI$
112Ø IF LEN(TI$)<>6 THEN445
1125 GOTO 445
113Ø REM ***** PRESENTACION HORA *****
1135 PRINT "[CTRL+6][CLR/HOME][ 22*CRSR ABAJO][ 40*COMM+@]";
```



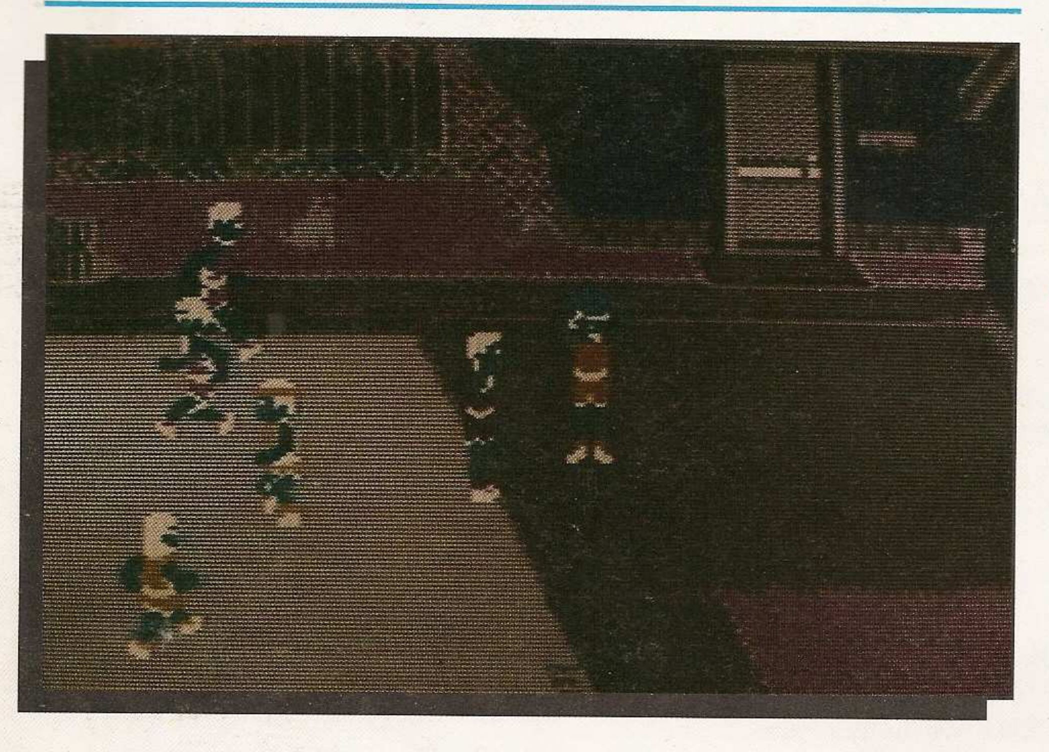
```
1140 PRINT "[SHIFT+H] => ";TI$
1145 FOR PR=0 TO 1000
1150 NEXT PR:GOTO 550
1155 REM ***** GUIA TELEFONICA *****
1160 PRINT "[SHIFT+CLR/HOME][ 10*CRSR ABAJO][ 9*CRSR DCHA.]";
1165 PRINT "[CTRL+6][SHIFT+L]ISTIN TELEFONICO":FOR A=0 TO 900:NEXT A
1170 PRINT "[SHIFT+CLR/HOME]"
1175 PRINT TAB(5);"[SHIFT+O]PCIONES":PRINT TAB(15);"========":PRINT "[ 2*CRSR ABAJO]"
1180 PRINT "[SHIFT+E]NTRAR NUEVO NUMERO TELEFONICO ======> 1"
1185 PRINT "[SHIFT+B]ORRAR NUMERO TELEFONICO[ 7 ESPACIOS] ======> 2"
1190 PRINT "[SHIFT+B]USCAR UN NUMERO EN EL ARCHIVO ======> 3"
1195 PRINT "[SHIFT+C]ARGAR EL ARCHIVO[ 14 ESPACIOS] ======> 4"
1200 PRINT "[SHIFT+S]ALVAR EL DIRECTORIO [ 11 ESPACIOS] ====== 5"
1205 PRINT "[SHIFT+L]ISTAR TODOS LOS NUMEROS[ 7 ESPACIOS] ======> 6"
1210 PRINT "[SHIFT+M]ENU PRINCIPAL[ 17 ESPACIOS] ======> 7"
1215 PRINT "[ 2 * CRSR ABAJO][ 40*COMM+@]";
1220 PRINT "[ 8 ESPACIOS][SHIFT+P]ULSE LA OPCION DESEADA"
1225 GET K$:IF K$=" " THEN 1225
1230 IF K$<"1" OR K$>"7" THEN 1225
1235 PRINT "[SHIFT+CLR/HOME]":ONVAL(K$)GOSUB1245,1285,1325,1530,1440,1380,550
1240 PRINT "[SHIFT+CLR/HOME]":GOTO 1170
1245 P=0:F=-1
1250 IF T$(P,0)=" " THENF=P:P=100
1255 P=P+1
1260 IF P<> 101 THEN 1250
```



```
1265 IF F=-1 THENPRINT "[SHIFT+M]EMORIA [SHIFT+L]LENA":FOR P=Ø TO 9ØØ:NEXT:RETURN
127Ø OPEN 1,Ø,Ø:PRINT "[SHIFT+N]OMBRE => ";:INPUT#1,T$(F,Ø):PRINT
1275 PRINT "[SHIFT+N]UMERO => ";:INPUT#1,T$(F,1):PRINT :CLOSE 1
128Ø RETURN
1285 OPEN 1,Ø,Ø:PRINT "[SHIFT+N]OMBRE A BORRAR";:INPUT#1,D$:PRINT :CLOSE 1
129Ø S=-1
1295 FOR X=Ø TO 1ØØ
13ØØ IF T$(X,Ø)=D$ THENS=X
13Ø5 NEXT X
131Ø IF S=-1 THENPRINT "[SHIFT+N]O HE ENCONTRADO ESE NOMBRE":GOTO 1285
1315 T$(S,Ø)="":T$(S,1)="  "
132Ø RETURN
1325 OPEN 1,Ø,Ø:PRINT "[SHIFT+N]OMBRE ";:INPUT#1,N$:CLOSE 1:PRINT
133Ø S=-1
1335 FOR X=Ø TO 1ØØ
134Ø IF T$(X,Ø)=N# THENS=X
1345 NEXT X
135Ø IF S=-1 THENPRINT "[SHIFT+N]O HE ENCONTRADO ESE NOMBRE":GOTO 1325
1355 PRINT "[SHIFT+N]OMBRE ";T$(S,Ø)
136Ø PRINT "[SHIFT+N]UMERO ";T$(S,1)
1365 PRINT :PRINT :PRINT "[SHIFT+P]RESIONE CUALQUIER TECLA"
137Ø GET K$:IF K$="  " THEN137Ø
1375 RETURN
138Ø INPUT "[SHIFT+C]OPIA A IMPRESORA";P$
1385 IF LEFT$(P$,1)="S" THENOPEN 4,4,7:CMD 4
139Ø FOR P=Ø TO 1ØØ
1395 GET K$:IF K$="H" THEN1395
14ØØ IF T$(P,Ø)="  " THEN142Ø
14Ø5 PRINT :PRINT T$(P,Ø)
141Ø PRINT T$(P,1):PRINT
1415 PRINT "=======================================";
142Ø NEXT P
1425 PRINT :PRINT "[ 8 ESPACIOS][SHIFT+P]RESIONE CUALQUIER TECLA"
143Ø GET K$:IF K$="  " THEN143Ø
1435 RETURN
144Ø REM ***** 'SAVE' PROGRAMAS *****
1445 OPEN 1,1,1,"FECOM C64"
145Ø FOR F=Ø TO 1ØØ
1455 FOR E=Ø TO 8:IF C$(F,E)"  " THENC$(F,E)="-"
146Ø PRINT#1,C$(F,E)
1465 NEXT E:NEXT F
147Ø FOR A=Ø TO 1ØØ
1475 FOR B=Ø TO 4:IF S$(A,B)="  " THENS$(A,B)="-"
148Ø PRINT#1,S$(A,B)
1485 NEXT B:NEXT A
149Ø FOR X=Ø TO 1ØØ
1495 FOR Y=Ø TO 1:IF T$(X,Y)="  " THENT$(X,Y)="-"
15ØØ PRINTZ#1,CO:FOR Z=1 TO CO:FOR ZZ=1 TO 4:PRINT#1,D$(ZZ,Z)
15Ø5 NEXT ZZ,Z
151Ø PRINT#1,T$(X,Y)
```

1515
1520
1525
1530
1535 END

NOVEDADES EPYX



BALONCESTO

Esta vez en directo desde SUNNYVALE (la sede en Estados Unidos de EPYX) nos llegan muestras de lo que serán sus dos próximos bombazos: el **STREET SPORTS BASKETBALL** (en castellano, «baloncesto») y **SUBBATTLE**, una auténtica batalla submarina que entusiasmará a todos los amantes de juegos de estrategia. En el caso del baloncesto las fotos que te ofrecemos son reveladoras: te verás transformado de golpe en una especie de **HARLEM GLOBE TROTTER**, ya que en todo momento el juego se desarrollará en pistas callejeras. Deberás elegir tres componentes de tu equipo y sobre todo uno de los magníficos campos callejeros de los alrededores... Ojo, porque cada campo tiene características diferentes. Corres el riesgo de hacerte graves lesiones en una de ellas, en la que una enorme mancha de aceite está debajo de la canasta. También deberás cuidar de que los pases que hagas no den contra el bordillo... Como era de esperar de EPYX, la biblioteca de jugadores es muy amplia y cada uno está dotado de habilidades diferentes en el juego. El realismo alcanza como siempre cotas más altas. Por ejemplo, si el partido es bueno, un gato aparecerá en una ventana y mirará el transcurrir de la prueba; pero, si baja el nivel, puede que empiece a bostezar o incluso que se baje de la ventana para ir a otras ocupaciones... impresionante.

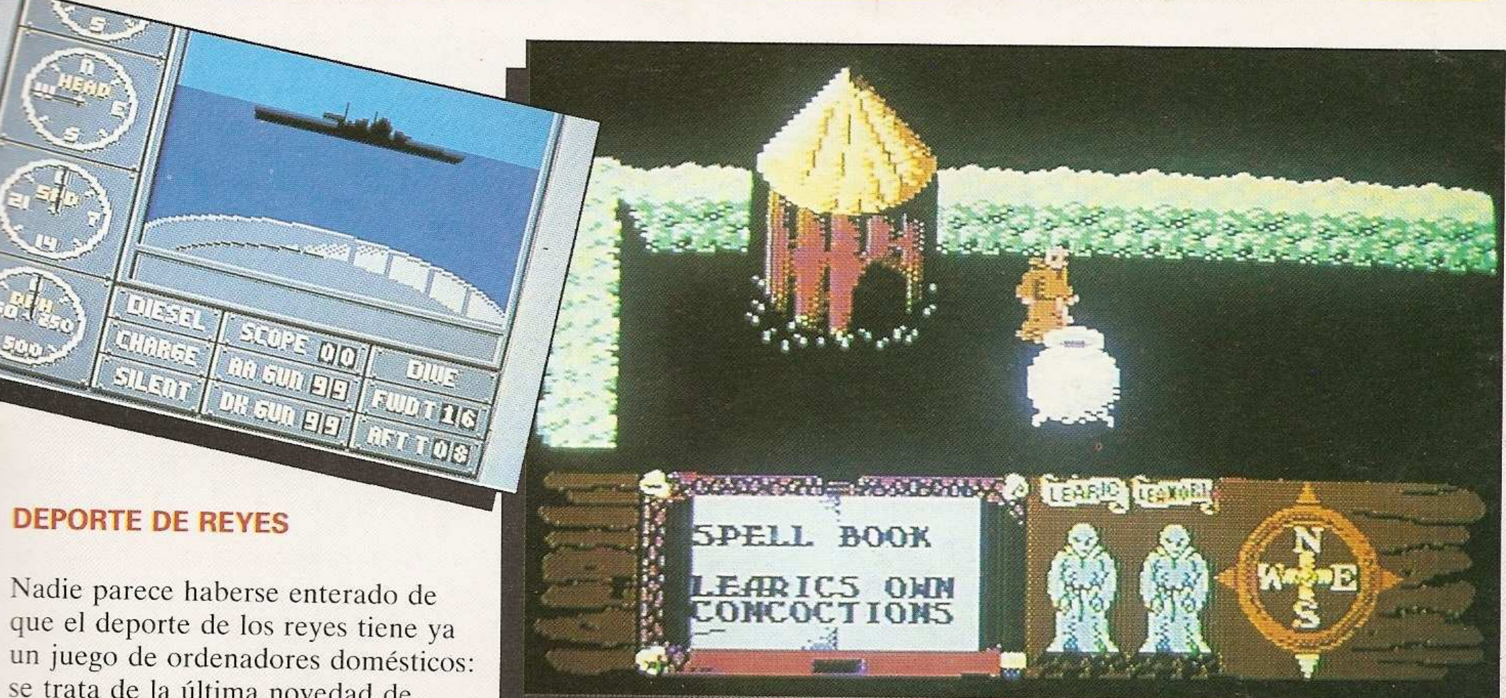


SUBBATTLE

En cuanto a la otra novedad, se trata de una fabulosa simulación de juego de submarino que te trasladará hasta la segunda guerra mundial... Los pocos que lo han visto aseguran que rompe con todo lo establecido en lo que a simulación se refiere.

PHM PEGASUS

Parece que **Lucasfilm** ha decidido dejar de estar con **EPYX**, o por lo menos eso es lo que parece tras la aparición del último título **—PHM Pegasus** el destructor— que esta vez se ha hecho en colaboración con **Electronic Arts**. Se trata de una simulación de guerra marina que se desarrolla en la segunda guerra mundial y tiene por escenario el mundo entero: desde el apresamiento y destrucción de una nave de piratas de los mares del sur hasta la captura de un submarino en aguas de Filipinas.



DEPORTE DE REYES

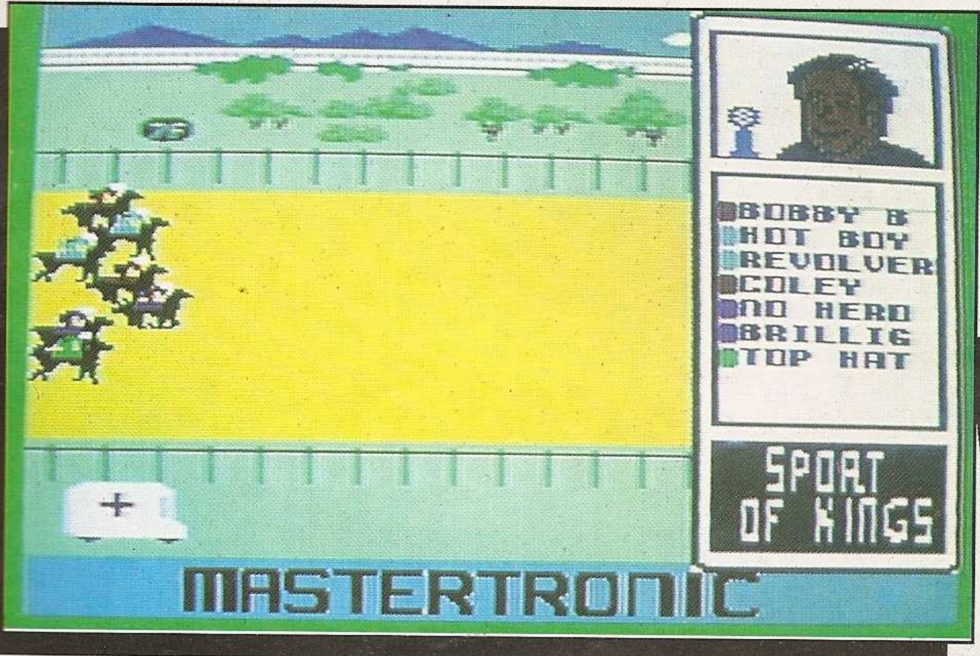
Nadie parece haberse enterado de que el deporte de los reyes tiene ya un juego de ordenadores domésticos: se trata de la última novedad de **Mastertronic**: un juego de carreras de caballos en la que podrás realizar las apuestas tan elevadas como jamás hayas soñado. El juego es de lo más completo e incluye desde un *book* de referencia de los caballos —en la que aparecen incluidos con todo el historial— hasta un ayudante de apuestas que te llevará al *paddock* a ver los animales antes de las carreras y te asesorará sobre la transcendencia del *handicap* de 12 kilos que le hayan dado al caballo que elegiste ganador o colocado. El juego incorpora también una serie de comandos que te permitirán crear tu propia base de datos.

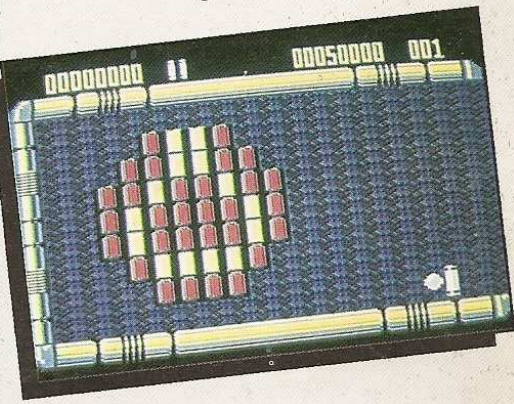
FEUD

La acción transcurre por tierras medievales en el siglo xv, cuando aún imperaban los magos, brujas y otros seres paranormales… un buen monje se dedica a vagar por tierras inglesas en busca de un manuscrito que le dará la clave de un tesoro que le hará inmensamente rico. El juego está lleno de colorido, las tierras están plagadas de brujas, magos y otros seres, los paisajes están muy diversificados y el juego está dotado de un sorprendente realismo gracias a la elección de los colores. Dicho realismo alcanza incluso la sofisticación de respetar el corte de los trajes de aquella época.

KRACKOUT

Este juego es el preferido por nuestra Redacción este mes, se trata de un plagio del **Arkanoid** (o viceversa), pero está dotado de elementos que lo hacen resaltar considerablemente frente a este último: velocidad de las bolas; multitud de efectos; rotación de fases; gráficos; opción de juego para zurdo y diestro; sonido; scrolling transparente y además todos los elementos que tanto nos gustaron de **Arkanoid**: Los ladrillos desprenden extraños objetos y van cambiando el juego… sin duda llegará a ser un clásico en su género.

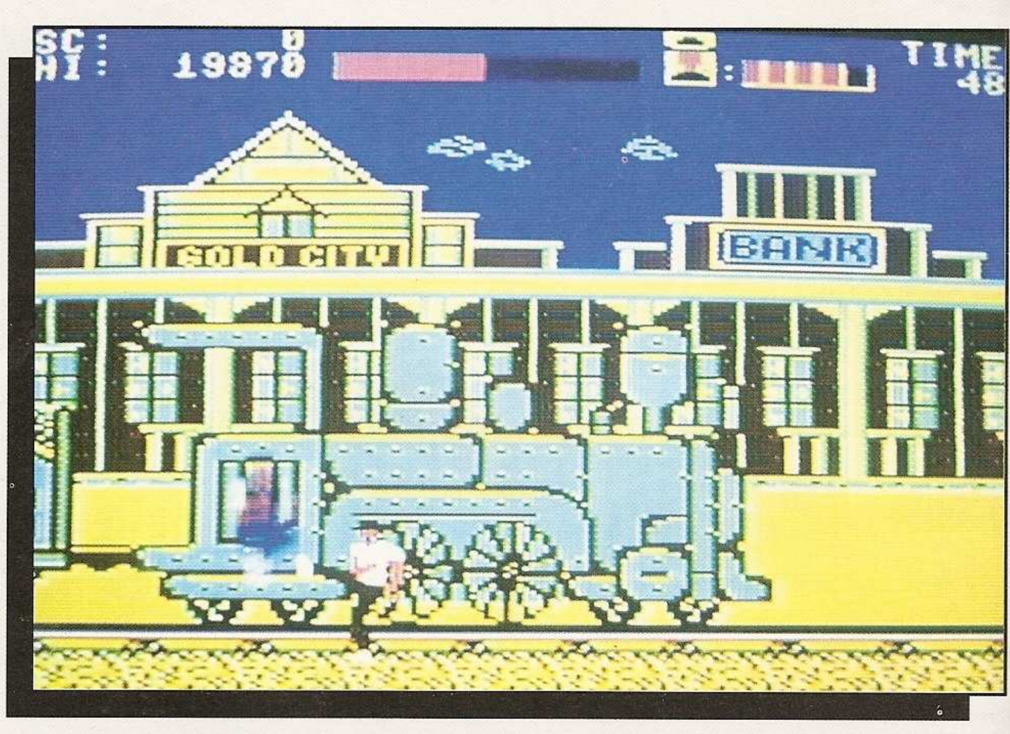




EXPRESS RAIDER

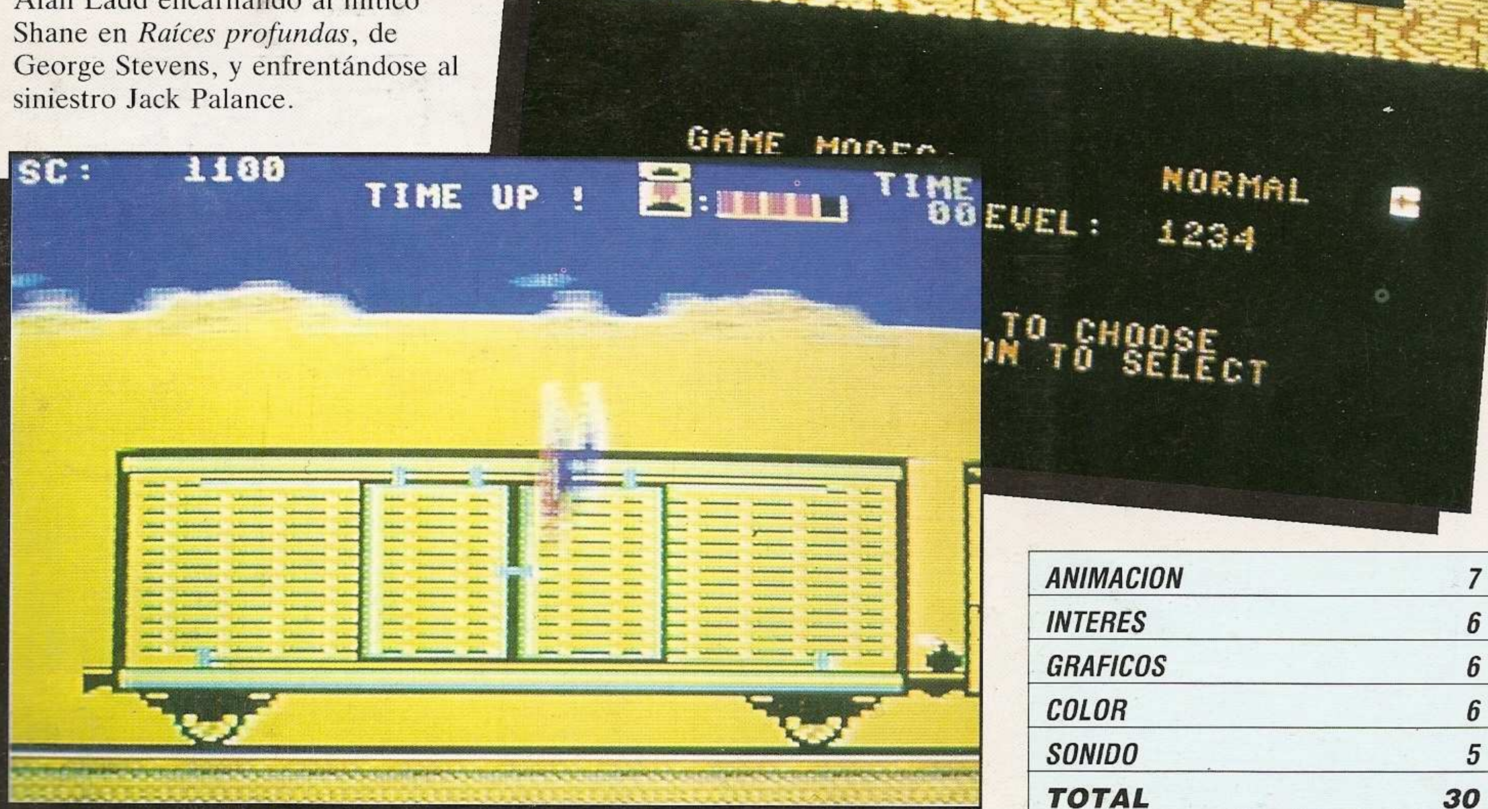
● ERBE ■ AVENTURA DEL OESTE

Los vaqueros eran aún más duros de lo que parece en las películas y, si no, ver lo difícil que es rescatar a las personas que se hallan en este tren. Deberás hacerte con el control del tren andando por el techo de los vagones. Verás cómo, enseguida, los techos se llenan de gente con cara de pocos amigos que te atizarán a la primera de cambio. Pero los peligros también vienen de los aires: numerosas aves pueden abatirse sobre ti causándote daños que harán bajar vertiginosamente tu indicador de fuerza. Otro de los peligros a los que deberás enfrentarte es la existencia de vallas colgadas en el camino recorrido por el tren: cualquier impacto con una de ellas te hará perder una vida sin ningún problema. La verdad es que la naturaleza ha dotado a nuestro héroe de singulares habilidades en la lucha: saltos a izquierda y derecha, que harían ponerse rojo de envidia a BRUCE LEE; tiros que no desmerecerían de los de JOHN WAYNE en la recordada película *La diligencia*, de John Ford, o los de Alan Ladd encarnando al mítico Shane en *Raíces profundas*, de George Stevens, y enfrentándose al siniestro Jack Palance.



Y, por supuesto, no debemos dejar de mencionar las patadas, que parecen de caballo...

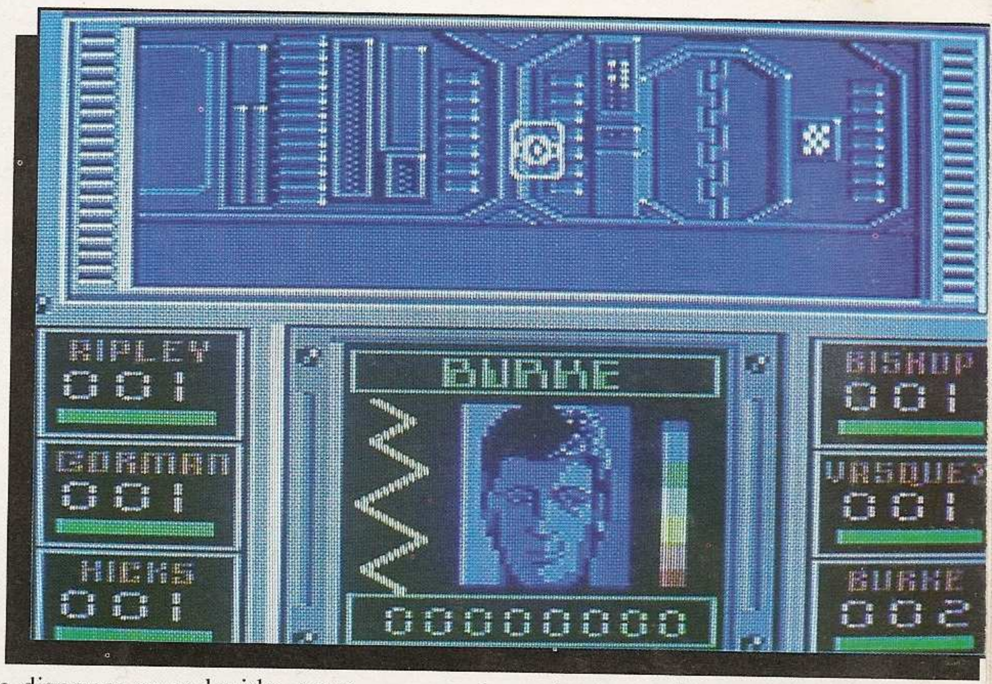
Desde luego el LEJANO OESTE era para gente realmente dura, y los alfeñiques no tenían nada que hacer allí.



ANIMACION	7
INTERES	6
GRAFICOS	6
COLOR	6
SONIDO	5
TOTAL	**30**

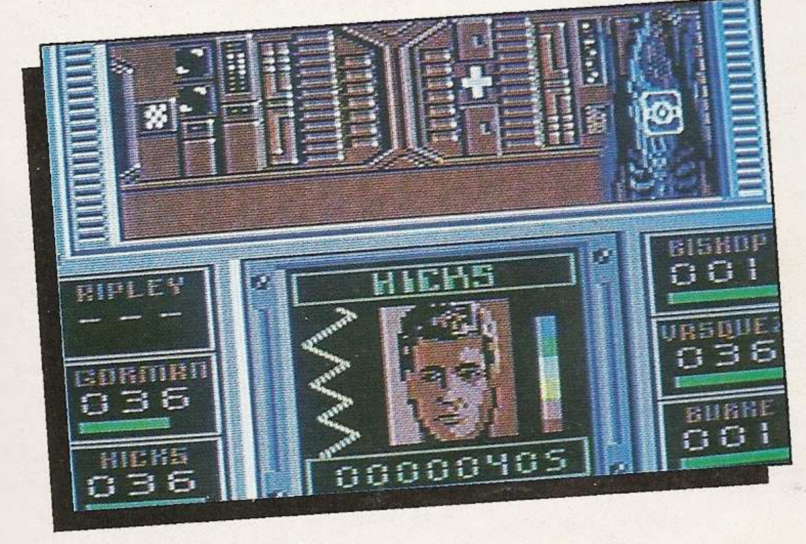
ALIENS

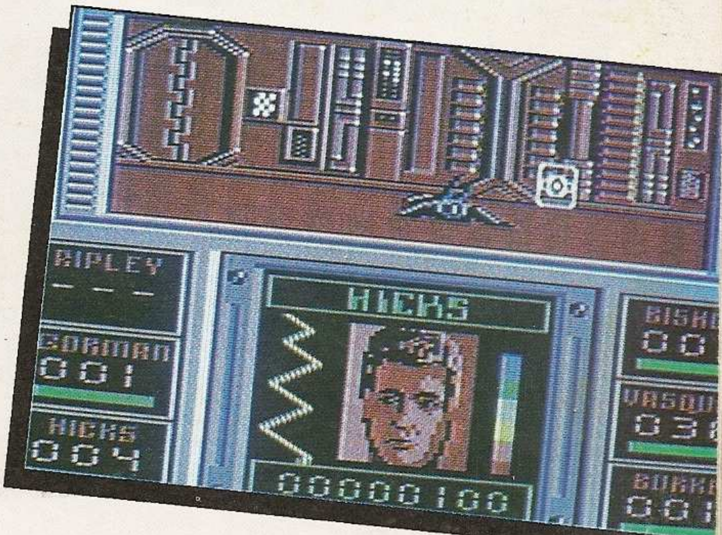
● PROINSA ■ JUEGO DE SALAS



Muchos han visto la primera película, titulada ALIEN (sin la letra S). La tripulación de la nave **NOSTROMO** está siendo diezmada por un bichejo con mala cara y mucho apetito que se acaba merendando a toda la tripulación, menos a una chica, **RILEY** (encarnada por la bella actriz norteamericana Sigourney Weaver en la versión cinematográfica).
La acción de ALIENS (EL REGRESO) se sitúa unos años más tarde. Llaman a RILEY para investigar un rumor que llega a la Tierra. Al parecer, centenares de bichos, como los que ella describió al volver de su periplo, han tomado una base en el espacio exterior. No vamos a contaros el estado en el que se halla el lugar: estos marranos lo han llenado de excrementos. Lo que sí podemos deciros es que son muchos, muy rápidos y muy listos. Debes destruir sus excrementos porque si no te cerrarán el paso de las puertas.
Mientras tú no puedes pasar por todas las puertas, ellos en cambio no tienen ese problema y a veces parece que son auténticos fantasmas por el modo de aparecer. Los *alienígenas* utilizan un curioso método para quitarte de en medio: la impregnación. Así, mientras uno te despista, el otro te impregna (verás el color de tu indicador cambiar hasta el rojo, que es igual a decir ¡ADIOS!). Como siempre, el equipo del que dispones es reducido, pero también debes saber que en las tareas de grupo lo que más importa es el elemento humano. Tienes un equipo de seis personas: **Riley**, que ya ha demostrado su temple; **HICKS**, quizás el mejor elemento del equipo (nervios de acero, rápido de decisión, lúcido como pocos...). El contrario de **HICKS** es **GORMAN**, el teniente, para el que ésta es la primera misión. También tenemos a **VAZQUEZ**, el soldado con mente fría y calculadora; por último, **BISHOP**, el oficial androide (no creáis ni por un momento que por ser androide no van a tratar de merendárselo los **ALIENS**). Para llevar a cabo la misión todos los miembros del equipo llevan el Mobile Tactical Operation Bay. Como es evidente, juegas a la vez con los seis personajes; en el MTOB verás representado el rostro del miembro de la tripulación con el que te halles; la habitación en la que está cada miembro, el estado físico (la longitud de la barra) y el psicológico o de impregnación (el color de la barra). Como los ALIENS entran y salen como Pedro por su casa, una alarma te indicará si en la habitación en la que estás hay o no ALIENS.
El juego viene acompañado de un mapa sin numerar.

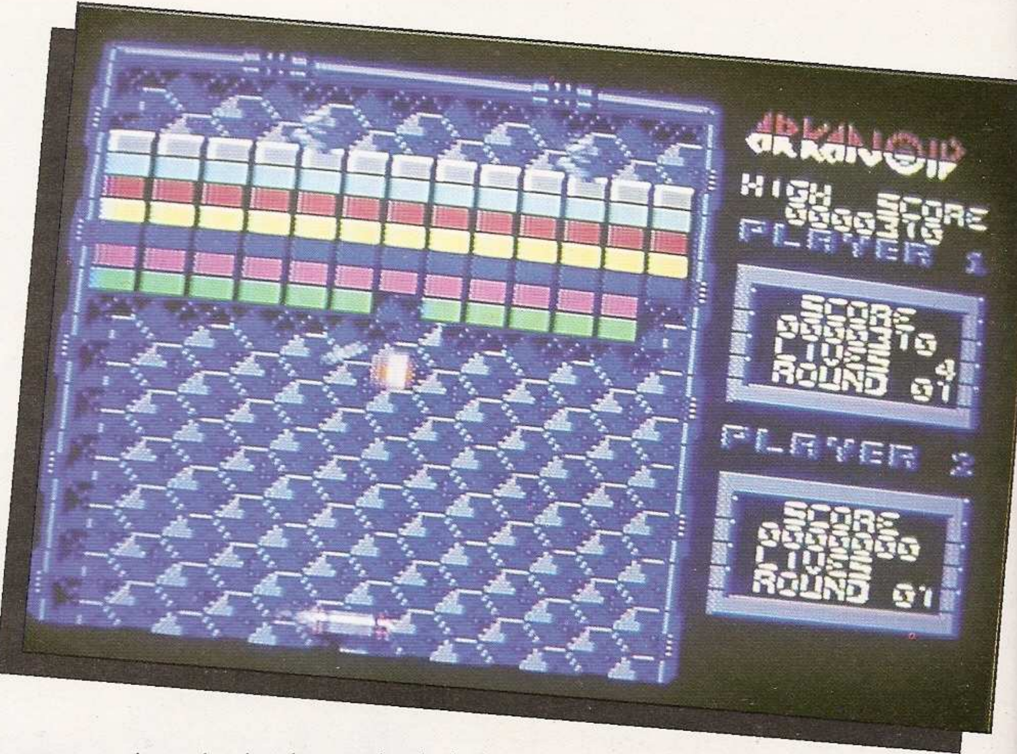
ANIMACION	9
INTERES	8
GRAFICOS	9
COLOR	8
SONIDO	5
TOTAL	**39**





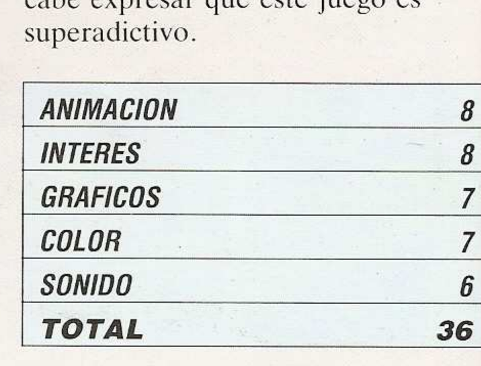
ARKANOID

● IMAGINE ■ JUEGO DE MUROS



Hasta que aparece la típica pantalla de presentación de una nave es probable que todos penséis que se trata de un juego de marcianos más. En cuanto deis a la barra espaciadora veréis que esto no es verdad, pues se trata de un nuevo universo en el que os va a ser muy difícil llegar a la última pantalla.
Arkanoid es otro tipo de clásicos: los de MUROS y BOLAS. Se trata básicamente de destruir barreras de ladrillos con una pala y una bola. Realmente el juego puede parecer soso hasta aquí: no os fiéis demasiado, ya que incluye una serie de variantes que aumentan notablemente su adictividad y lo hacen sólo apto para los virtuosos de la palanca. Conforme vayas destruyendo barreras verás como a menudo salen objetos de los lugares que ocupaban los ladrillos: el color de los objetos es lo que determina lo que va a suceder cuando te acerques a ellos.
Puede que temporalmente te halles inmovilizado y por tanto veas cómo tu escurridiza bola desaparece por el fondo, para quitarte una vida más, o bien que tu raqueta se duplica de tamaño, o también que se convierte en pegajosa, haciendo que las bolas se te queden pegadas y permitiéndote más fácilmente apuntar para destruir zonas difíciles.
En la parte superior de la pantalla verás también cómo aparecen unas extrañas naves que, a menudo, vendrán a romper los planes que hagas acerca de la trayectoria de las bolas, ya que la colisión de la bola con las naves cambia por completo la orientación de la pelota.
El juego permite todos los efectos posibles y te recomendamos mucho que uses y abuses de ellos ya que, a menudo, al hacer incidir la bola en diagonal lograrás destruir dos ladrillos a la vez (o a veces muchos más si se mete en un hueco).
Mientras que si los tiros son horizontales desaparecerá simplemente el primero de los ladrillos. Otra recomendación es que te cuides de *trabajar* una zona del muro. Es muy importante para que así veas cómo se puede tumbar casi enteramente un muro dejando a tu traviesa bola botar entre el frontis y la parte superior.
El juego incorpora también una variante peculiar al existir en cada pantalla un objeto que cae y te abre una trampilla en la pared lateral por la que accedes a la pantalla siguiente sin tener que acabar la anterior.
Como conclusión, solamente nos cabe expresar que este juego es superadictivo.





ANIMACION	8
INTERES	8
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	**36**

SHORT CIRCUIT

● ERBE ■ JUEGO DE SALAS

Tras una exhibición de las capacidades de los robots aplicados a las artes militares, un robot de la serie SAINT queda parcialmente afectado por un rayo de tal modo que el robot cobra misteriosamente vida propia y huye para salvar sus chips de los científicos.

NUMERO CINCO —ya que así es como se identifica al robot— está acosado por tres peligros. Uno, los militares que lo quieren eliminar por el considerablemente valioso y sofisticado armamento que lleva a cuestas, el segundo, los científicos, que como hemos dicho lo quieren *limpiar* para descubrir el milagro que se ha operado en la máquina, y, por último, los demás robots que tienen orden de apresarle o aniquilarle y disponen de un auténtico arsenal para llevar a cabo tan fatídica misión.

Nuestro amigo **NUMERO CINCO** precisa de tres elementos para poder salir del edificio: el *hardware* correspondiente al **RAYO LASER**, el *hardware* correspondiente al salto y el manual técnico que le permite crear un nuevo e idéntico **SAINT**.

El edificio en el que se hallan está plagado de sistemas de seguridad cada cual más sofisticado pero nuestro simpático robot es muy listo y tiene acceso a las computadoras de **NOVA**.

Dentro de la computadora se hallan tres tipos de programas: uno de búsqueda, otro de salto y otro de láser. Estos dos últimos deberán estar activados para poder funcionar y por otra parte la memoria de **NUMERO CINCO** solamente permite cargar dos programas a la vez, así que, como comprenderéis, debéis de cargar en primer lugar el programa de **SEARCH** (búsqueda). En lo que al *hardware* se refiere, **NUMERO CINCO** únicamente puede llevar tres objetos a la vez, así que ojo con lo que lleváis. Los objetos son fáciles



de hallar y el único problema posible está en los que se hallan dentro de los ficheros. Dicho sea de paso, casi siempre los objetos más importantes se encuentran en el interior de ficheros.

El juego está plagado de **PASSCARD** (tarjetas de pases), **PASSWORD** (palabras claves) que te permitirán acceder a la mayoría de las salas, para otras necesitarás un manojo de llaves **(BUNCH OF KEYS)**. EL **PASSWORD** te servirá para poder entrar en la zona más protegida del edificio y para apagar y encender el ordenador pero, ¡ojo!, cada ocho minutos deberás volver a encenderlo, ya que se produce un cambio de guardia y se darán cuenta de que el ordenador central ha sido manipulado.



El juego tiene una excelente combinación de colores en las pantallas que corresponden a la parte del exterior de la factoría. Es muy agradable y resulta fácil elaborar un mapa.

En resumen, **CORTOCIRCUITO** tiene todo lo que se precisa para constituirse en un auténtico éxito.

ANIMACION	9
INTERES	8
GRAFICOS	9
COLOR	10
SONIDO	6
TOTAL	**42**

THE GOONIES

● **ERBE** ■ **ARCADE**

MAMA FRATELI no sabe el tesoro que tiene debajo de sus pies, tampoco andan del todo seguros nuestros amigos los **GOONIES**, que parece que andan un poco despistados y que para salvarse de la *señora gorda* deben colarse en la chimenea. Éste ha sido un juego precursor en lo que se refiere a posibilidad de jugar a dos ayudándose uno con otro y siendo cada jugador autónomo.
El juego se basa constantemente con la ayuda de los dos jugadores y obliga a menudo a relizar un esfuerzo intelectual: os contamos cómo alcanzar y superar las pantallas 7 y 8, que a nuestro entender son las más difíciles.
Los personajes están muy bien caracterizados. Queda patente la avaricia de la **MAMA**, avaricia que a menudo servirá para mantenerla entretenida mientras uno de los nuestros hace de las suyas. No hace falta deciros que el juego tiene ocho fases por las que pasarás entre peligros de lo más diverso: desde el riesgo de ahogarse, ser aniquilado por un pulpo —esperamos que así no sea, ya que os damos la solución de esta fase— a ser machacado a porrazos por la MAMA o por cualquiera de sus hijos.
El feliz desenlace es la pasta que se llevan al bolsillo nuestros amigos ante la mirada atónita de los adversarios.
Un juego imprescindible en cualquier ludoteca que se precie y que hará las delicias de los que no lo conocían antes.

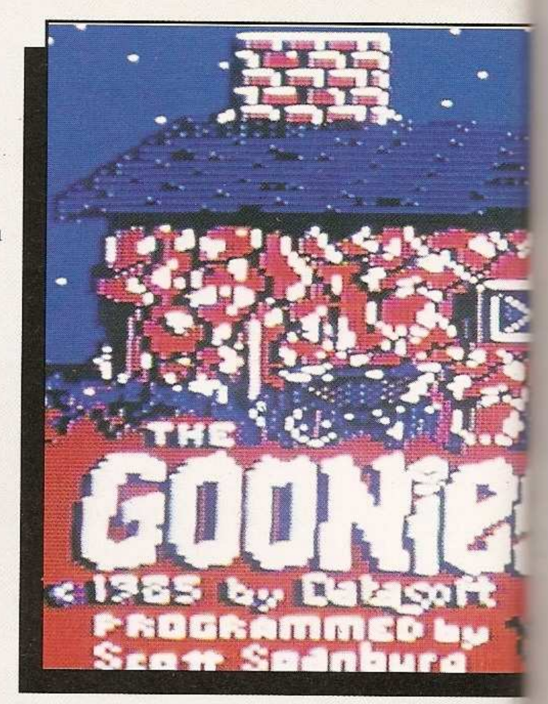
TRUCOS

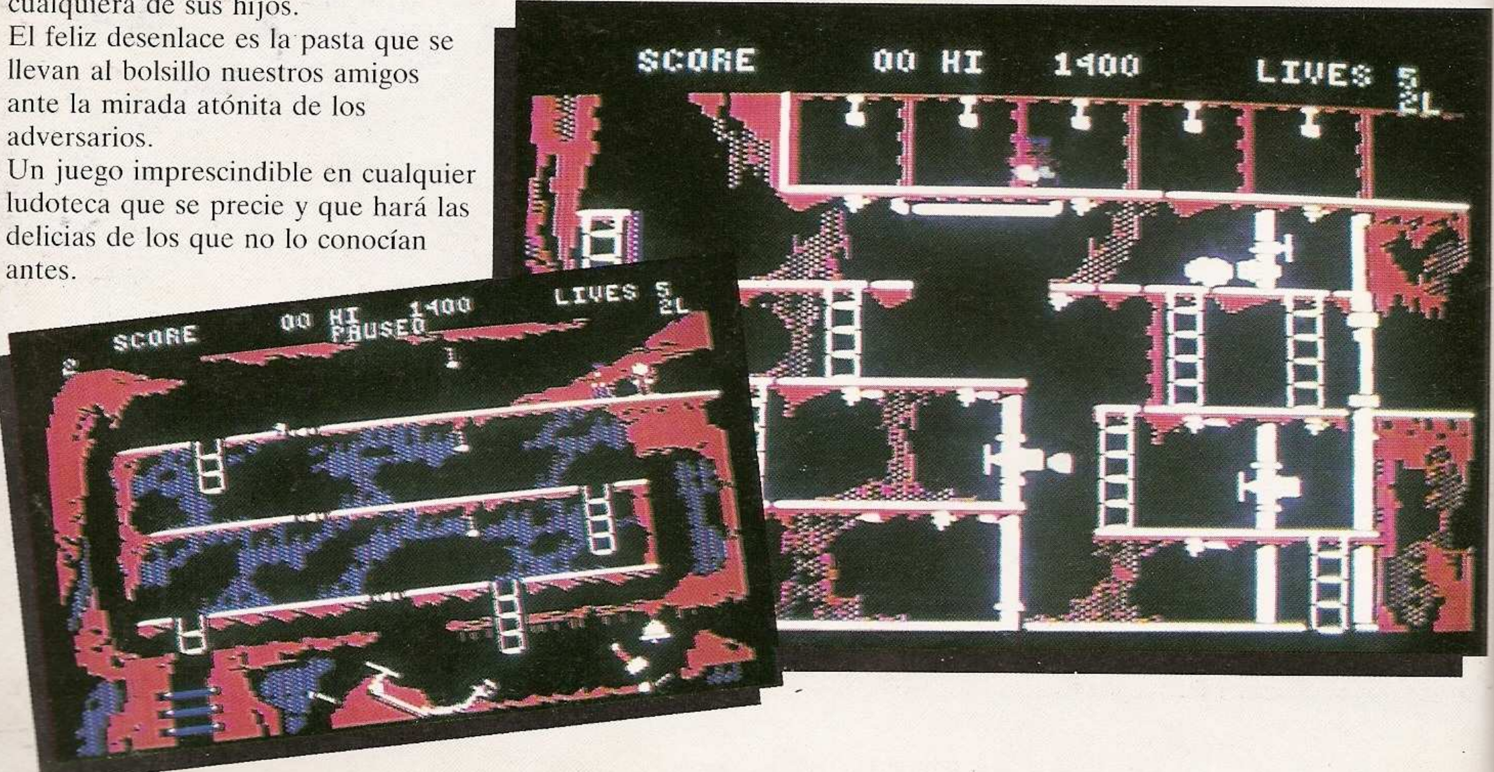
PANTALLA NUMERO 7

1. Los *GOONIES* van a la escalera.
2. Uno se va arriba a la izquierda y, gracias al trampolín, salta y se agarra a la cadena arriba a la izquierda, mientras que abajo una caja cae al suelo.
3. Uno de los *GOONIES* se queda en la escalera, el otro se va abajo. Debe empujar la caja hasta la altura de la cadena y muy deprisa saltar sobre la misma para agarrarse a la cadena.
4. El que se ha quedado en la escalera sube, salta sobre el trampolín lo más a la izquierda posible, para quedar cerca del pulpo. A estas alturas el nivel del agua ya ha bajado; puede saltar encima del pulpo sin morir. El otro *GOONIE* puede bajar por la cadena y va hasta su compadre... Pasan entonces a la fase 8.

PANTALLA NUMERO 8

1. El más pequeño va a la palanca que está junto a la **MAMA**.
2. Esta palanca abre el pasadizo. El grande va a accionar la cadena en el medio a la izquierda.
3. El pequeño sube y pasa por el pasadizo para accionar la cadena del medio-izquierda.
4. El grande baja y acciona la cadena de abajo-izquierda.
5. El pequeño baja.
6. El grande sube hasta el cofre arriba a la derecha y lo empuja al mar. El dinero se hunde y la **MAMA** se ahoga.
7. El peque acciona la cadena de abajo-izquierda.
8. El grande va hasta la palanca que







había donde estaba la **MAMA** y *NO LA TOCA*.
9. El peque va al montacargas, del medio-arriba.
10. El grande acciona la palanca.
11. El peque salta a la derecha y va hasta el fondo.
12. El grande le alcanza.
13. **Te has convertido en un GOONIE.**

The *Goonies* —juego inspirado en la exitosa película del mismo título producida por el famoso «mago del cine» Steven Spielberg— es el típico programa impredecible e imprescindible que le hace a uno sentirse protagonista de hazañas más bien propias del mismísimo Indiana Jones.
En efecto, al introducirte en su trama, recreas las situaciones de peligro, te conviertes en héroe «caballeresco-andante» sin caballo, luchas contra sórdidas e ignotas legiones de fuerzas del mal, juegas al gato y al ratón con tus enemigos...
Estas pueden ser algunas de las razones y emociones que justifiquen la visualización de este interesante arcade de aventuras de películas. Recordemos la trama de esta aventura.

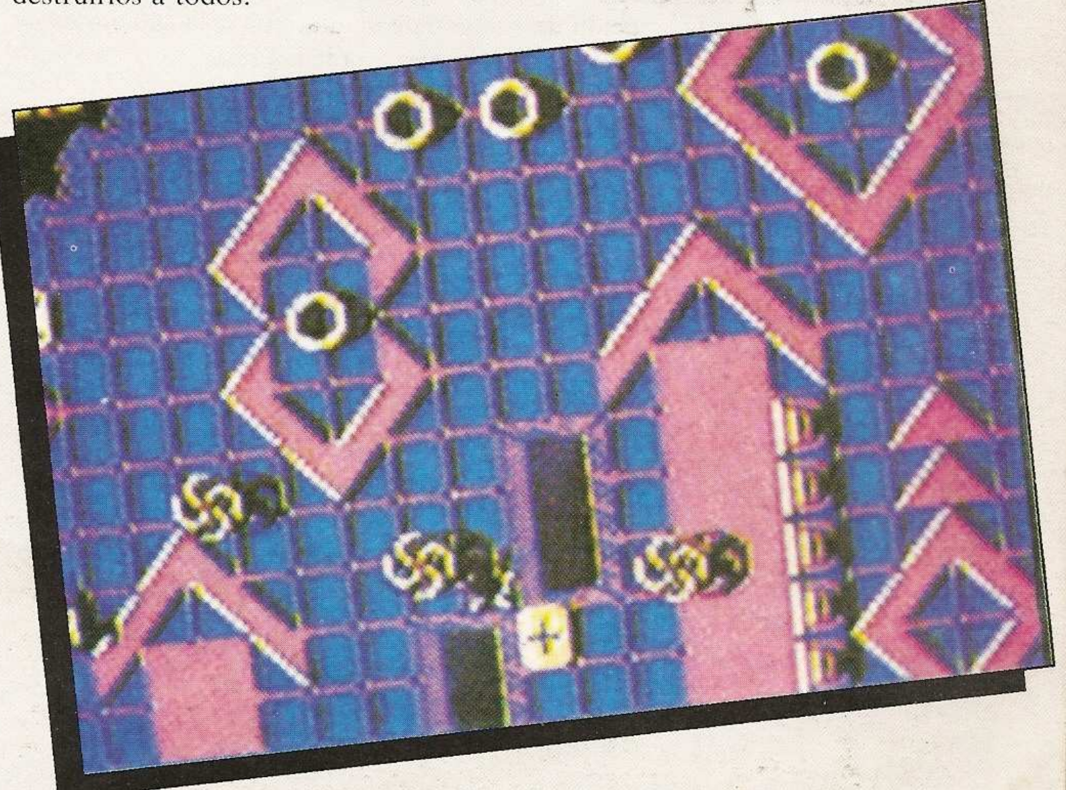
ANIMACION	8
INTERES	10
GRAFICO	8
COLOR	8
SONIDO	7
TOTAL	41

FIRETRACK

● **PROINSA** ■ **ELECTRIC DREAMS**

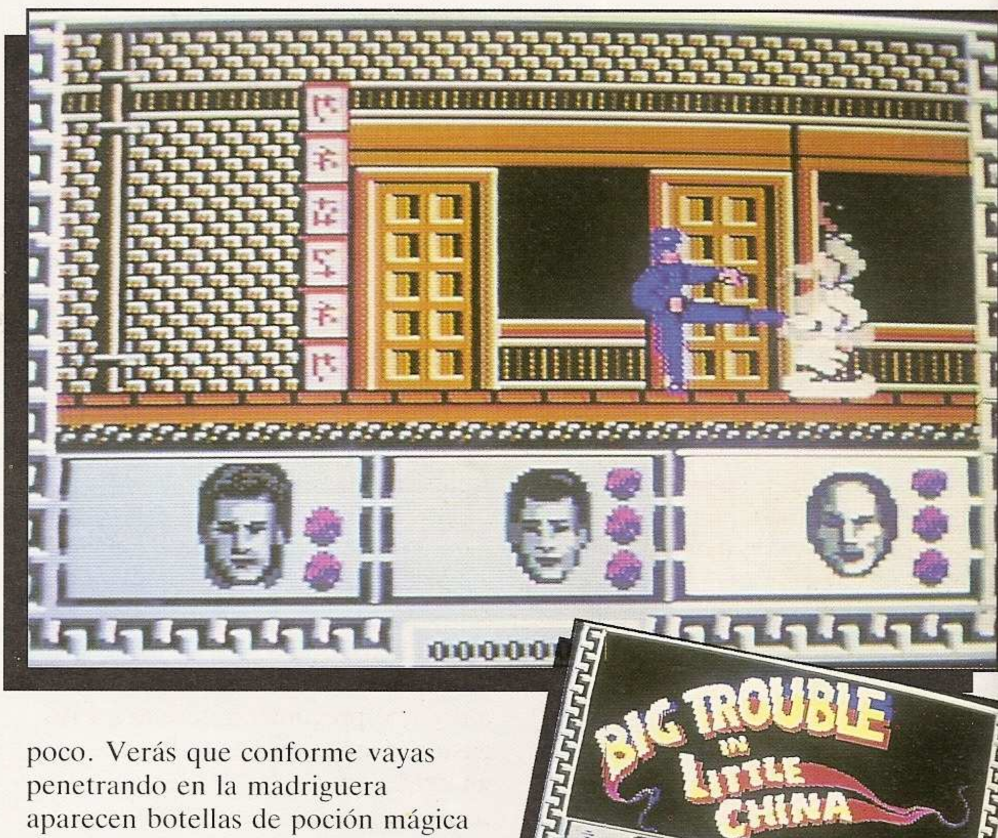
Los juegos de marcianos siempre tienen unos escenarios raros, planetas lejanos, civilizaciones perdidas y amenazadas. A menudo muy buen *instrumental* para poder defenderse y repeler el ataque de los *malos*. No vamos a contaros esta vez la película que hay alrededor de esta nueva STARWAR. Como ya os suponéis debéis de pegar tiros a todo lo que se mueva delante vuestro, la cosa va de tiros y deberéis dar muchos más de los que estáis acostumbrados si queréis salir ilesos de este recién salido juego de marcianos... Los gráficos son buenos y quizás lo que más resalte sean las raras y *alienígenas* combinaciones de colores del planeta en el que se están viviendo tantos conflictos... Los ataques a los que te verás expuesto cobran aires y pinta de lo más raros y extraños: a veces serán sólo naves sueltas o bien podrán ser auténticas colmenas de naves con forma de caracol que se te echarán encima y que al recibir tu disparo se convertirán en mil y una bolas mortíferas.
A los que vayan a intentar la aventura de acabar este juego, la única recomendación que se les puede hacer es que se queden en el medio de la pantalla. En efecto, el juego permite una mobilidad hacia la derecha, izquierda, adelante, esto te será vital cuando te vengan varios enemigos a la vez, ya que mientras disparas podrás ir echándote hacia atrás ganando así tiempo para destruirlos a todos.

ANIMACION	7
INTERES	6
GRAFICOS	7
COLOR	7
SONIDO	5
TOTAL	32

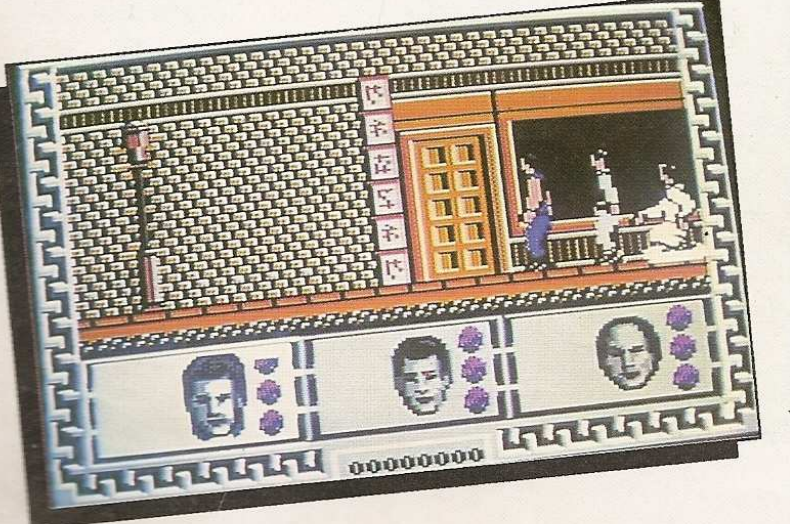


GOLPE EN LA PEQUEÑA CHINA

• PROINSA ■ JUEGO DE SALAS



La historia no es muy sencilla: un mandarín llamado **LO PAN** está intentando calmar a un demonio, y para ello precisa de un sacrificio: debe casarse con una joven de ojos verdes y posteriormente sacrificarla. Las novias de nuestros dos héroes (en la película **Jack Burton** y **Miao Yin**) son raptadas para tal fin y llevadas al imperio del mandarín **LO PAN**, que se halla debajo del barrio chino de **SAN FRANCISCO**.
Aquí es donde empieza el juego: JACK, WANG y otro acompañante llamado **EGG** penetran en el universo de **LO PAN** para rescatar a sus amigas. Durante el juego vas a controlar los tres personajes. Cada uno tendrá una serie de habilidades que tendrás que saber utilizar en cada momento. **Jack** luchará al principio sólo con los puños, aunque luego si llegas suficientemente hondo en el muno de **LO PAN** encontrarás una pistola con munición casi ilimitada esparcida a su alrededor.
WANG es un experto en ARTES MARCIALES y hará las delicias de todos los amantes de juegos de porrazos: domina todos los golpes de las artes y de vez en cuando verás que maneja las espadas que se encuentra en el camino con la maestría de los míticos **NINJAS**.
EGG, el último de los aventureros, es un mago poderoso que hace brotar de sus dedos rayos mortíferos, cuyo único defecto es que duran muy poco. Verás que conforme vayas penetrando en la madriguera aparecen botellas de poción mágica que refuerzan su poder o alargan la duración de sus rayos.
En la parte inferior de la pantalla verás un indicador de estado de los tres personajes. Al ser un programa con connotaciones de tipo ORIENTAL el indicador de fuerza viene dado por el signo del **YING-YANG**.
El juego se desarrollará en 4 niveles cada cual más difícil que el anterior: el primero se ubica en las calles de **CHINATOWN:** en este nivel muy pocas veces encontrarás pistoleros, casi siempre serán luchadores de artes marciales a los que deberás plantar cara.
Cuando te cruces con un pistolero, echa a correr ya que no te queda más opción.
En el segundo nivel, ya en las alcantarillas, deberás luchar contra monstruos. Te recomendamos que saltes por encima de ellos ya que, como verás si lo intentas, son imposibles de ganar en combate.
Ya en el cuartel de **LO PAN** deberás derrotar a extraños seres que llevan sombreros de ala ancha antes de poder pretender alcanzar al dueño del lugar. Si consigues esto quizás puedas penetrar en la habitación donde se prepara la boda. En ella verás cómo **LO PAN** se halla en suspensión en una nube. Derrotar sus guardaespaldas será la última prueba antes de la lucha cuerpo a cuerpo con él. La lucha con él es la prueba más difícil, ya que **LO PAN** tiene poderes sobrenaturales que le permiten regenerarse como el **AVE FENIX**.



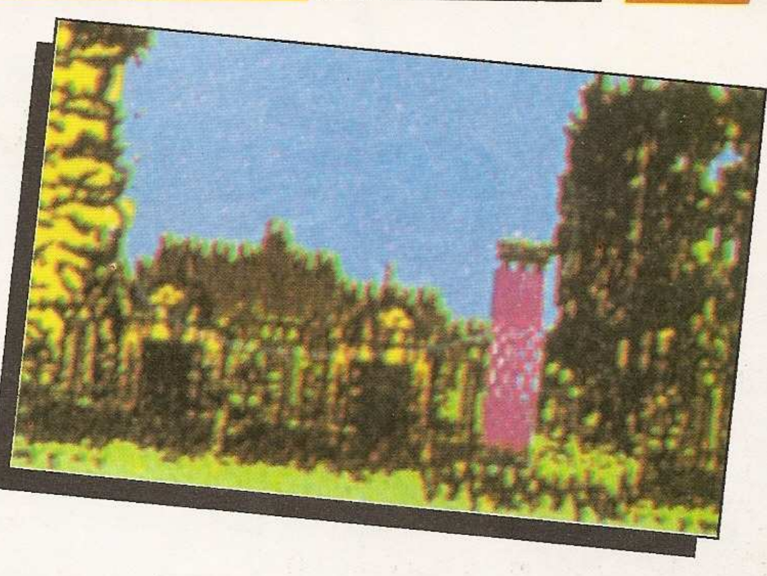
ANIMACION	*9*
INTERES	*9*
GRAFICOS	*10*
COLOR	*9*
SONIDO	*8*
TOTAL	***45***

EXPLORER

● PROINSA ■ ELECTRIC DREAMS

Hay hombres que tienen muy poca fortuna: baste como ejemplo nombrar el caso del héroe de este juego. Haceos a la idea de que se ha quedado más solo que la una en un hostil páramo del planeta ESMERALDA. Todo por no comprar una aeronave nueva... claro que, sin duda, es ya mucha suerte el haber quedado con vida y todavía le queda una oportunidad de salvarse: ... buscar y encontrar los nueve trozos en que la nave ha quedado segmentada tras el accidente. Donde realmente empieza la mala pata de nuestro protagonista es aquí, porque el planeta tiene 40 billones de lugares en los que pueden hallarse los restos de la nave.

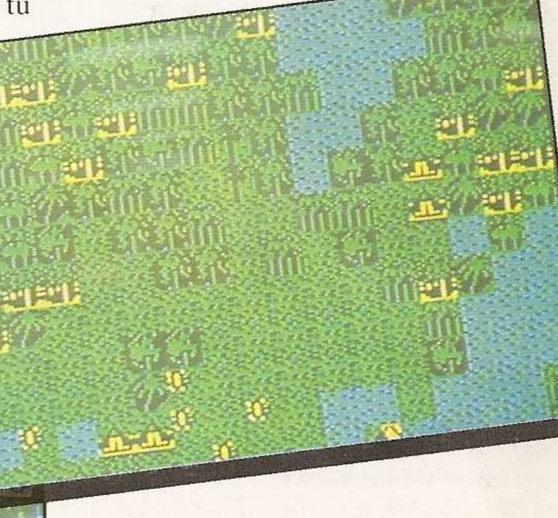
El juego gana en complejidad a partir del momento en que comienza la exploración. Un potente y sofisticado equipo te ayudará a encontrar los restos de tu aeronave: un jet portátil te ayudará en tus desplazamientos; un sonar de objetos te ayudará a localizar los segmentos utilizando la onda de refracción de los mismos y averiguar la distancia aproximativa a la que se hallan. El radio vector, la brújula, la pistola láser y las botas de monte te serán también sin duda imprescindibles para andar por la jungla de este lejano planeta llamado ESMERALDA, que constituye apenas un punto perdido en medio de la inmensidad del universo sideral.



Verdad es que este planeta desafía cualquier inteligencia conocida. A menudo deberás utilizar radiofaros para guiarte por el llamado método de triangulación. Procura andar con cautela por el planeta, ya que descubrirás a menudo oscuras manchas en la selva que tras hacerte mil y una preguntas referentes a tus gustos, sueños, etc..., te llevarán a un extraño lugar: ¡OJO!!, porque como buen explorador —descubridor de tierras lejanas— deberás darle tú un nombre al lugar.

Procura andar con sumo cuidado al realizar este «bautizo» ya que puede que te armes un auténtico follón en cuanto te teletrasporten cuatro o cinco veces.

Quizás lo más sorprendente del juego sea el escrupuloso cuidado con que se han tratado los gráficos, se ha logrado crear un curioso efecto de profundidad por medio de los *FILLS* (rellenos). En los lugares por los que irás vagando podrás utilizar un *zoom* para buscar mejor las nueve piezas...



Se trata, como puedes ver, de un programa superlativo. Así que, como un buen Robinsón Crusoe sideral, anímate y empieza cuanto antes la búsqueda, ya que este juego sin duda se lo merece plenamente.

ANIMACION	9
INTERES	10
GRAFICOS	10
COLOR	9
SONIDO	8
TOTAL	**46**

EL ZOCO

Deseo contactar con usuarios del C-64 para intercambio de programas últimas novedades. Máxima fiabilidad en las reproducciones. Rafael Navarrete Ruiz. C/ Bordadoras, 3 bajo-C. 41008 Sevilla.

Vendo unidad 1541, con un año de uso aprox. en perfecto estado por cambio a una unidad superior, por 40.000 pts. Regalo los siguientes programas: Super base 64, Easy script y una hoja de cálculo, así como otras utilidades y algunos juegos. Interesados remitir oferta a: Fco. Javier Salcedo Sierra. C/ Alto de San Isidro s/n. 49002 Zamora Tl: (988)527762.

Vendo ordenador C-64 más unidad de cassette C2N más un montón de revistas y juegos, todo al precio de 49.000 pts. Vendo también ensamblador llamado HESMON a 3.000 pts. y cartucho Simon's Basic a 6.000 pts. Miguel Tallón Biarge. C/ Ribera del Beiro bl. 7 3.°B. 18013 Granada.

Intercambio juegos en cinta para C-64, todos ellos en turbo. Me interesa el Simon's Basic. Mandar lista: Juan C. López. Pgno/La Paz, 2, 6.°, B. Gurutzetak-Baracaldo. Bizkaia 48900.

Intercambio programas en disco para el C-64, poseo sobre 1.000 (entre las cuales últimas novedades), se asegura seriedad. Mandar lista. Antonio G. Rodríguez. A. Postal: 3032. 36200 Vigo.

Cambio todo tipo de programas (juegos y utilidades) en cinta para C-64. Si alguien sabe jugar o tiene instrucciones de DRUIDS; desearía cambiárselas o comprárselas. Óscar García González. C/ Olmedo, 40 1.°F. Valladolid 47013.

Intercambio programas para el C-64. Todos ellos en turbo. Poseo más de 500. Alvaro Ginel Díez. C/ Arca Real, 1, 3.°, D. 47013 Valladolid.

Vendo C-128-D por cambio a PC. Además incluyo: instrucciones de gran cantidad de juegos. Abundantes revistas sobre Commodore. Curso de programación (BASIC BASICO). Cuatro tomos de Informática Aplicada, 8 primeros tomos de taller de Informática. Manual del usuario. Manual de la unidad de discos 1571. Cuatro cassettes con programas. Dos joystick Quick Shot II. Y otras cosas más. Manuel Jiménez Jiménez. C/ Alcalde Ortega Muñoz, 6, 2.° izq. Cp: 29014. Málaga. Tl: 262220.

Vendo CBM-64, Datacassette, 200 programas en cinta, 25 revistas, Interface para hacer copias de seguridad entre datacassettes. Unidad de discos 1.571 con garantía, 50 discos con 300 programas, el libro CBM 1571-1570. El gran libro Floppy de data-Becker. Y otras cosas más. Precio a convenir. Llamar a Paco de Barcelona. Tl: 2138549.

Vendo equipo completo compuesto por C-64, datacassette C2N, libros y manuales en español , guía de referencia, interface copiador, joystick Quickshot II, programas Logo, Pascal, Simon's Basic II y juegos originales. También incluyo disco de demostración y utilidades. Negociable. Jordi Aljama Deulofeu. C/ Del Comte Borrell, 4, Baixos 1.ª. Sant Celoni. Barcelona 08470.

Cambio programas de todo tipo en cinta o disco para el C-64. Poseo buenos y novedosos títulos. Escribir a : Vicenç Vila i Tomasa. C/ Socitat, 28, 4.°, C. Terrassa. Barcelona.

Vendo juegos para Commodore, últimas novedades. También compro los que me puedan interesar. Compro unidad de disco 1541 en buen estado a cambio de una larga lista de programas (poseo más de 1.000) abonando, si es necesario, 10.000 pts. Tl: 4229736 de Barcelona. Isidro Morón.

Cambio programas para el C-64 en cinta como: Simon's Basic, Gauntlet, Uridium, Commando, Infiltrator... por programas como: Acrojet, Pascal, Ace of Aces, Silent Service... etc. Enviar lista a: D. Almagro, Plaza Guillermo, 4, 4.°, 1.ª. Andorra la Vella. P. de Andorra.

Necesito instrucciones juegos C-64. Envia lista, precio y teléfono. Contesto todas las cartas y ofertas. Pref. Barcelona. También compro cartuchos utilidades, tipo freezer e interface MIDI.G.D. Apdo. 93045 Barcelona 08080. Tl: (93) 2179080.

Compro Impresora para C-64 directa y por fricción, por precio módico. También intercambio programas, enviar lista y ofertas a Miguel Ángel Gallardo García. El Drago bl. 11, 3.°, B. 29017 Málaga.

Cambio juegos para C-64. Interesados llamar o escribir: Santiago Sebastián. C/ Mayor, 52 1.°, D. Torres de la Alameda. Madrid. Tl: (91) 8858164.

Vendo C-128 cassette C2N, Joystick spectravideo, 13 cintas de juegos, 40 revistas de Informática, 3 libros editorial DATA-BECKER para C-64/128, programas de utilidades, Disco sistema Op. CPM PLUS, Simon's Basic. Javier Antón Morán. C/ El Milagro, 26. Torrelavega. Cantabria. Tl: (942) 881254.

Cambio programas para el C-64 en cinta. Tengo novedades en juegos y todos son de calidad. También tengo utilidades. Jorge Arias Carrera. C/ Río Valcárcel, 1, 5.°, B. 24400 Ponferrada. León. Tl: (987) 401258.

Vendo/cambio programas para C-64. Poseo alrededor de 700. Tengo unidad de disco y cassette. Fco. Javier Camacho García. Bl. 12D, 8.°, 6.°. Algeciras. Cádiz. Tl: (956) 664414.

Cambio programas por HARDWARE, interesándome especialmente C-64/1541, aunque aceptaría todo tipo de aparatos de cualquier marca. Desde ordenadores a transformadores pasando por consolas. Cantidad de títulos. A convenir. No importa el estado. También vendo los juegos. Marcos Miguel Huerga Muñoz. C/ Candelaria Ruiz del árbol, 55 4.°, p 3.ª A. 49003 Zamora. Tl: (988) 528864.

Vendemos juegos para el C-64, a buen precio, aprox. 500 pts. c/u. Tales como: Rambo, Baloncesto, Terra Cresta, Speed King y muchos más. Poseemos más de 50 juegos diferentes. Llamar a estos teléfonos: (93) 4218145 Javi o (93) 3338683 José.

Intercambio programas de juegos y utilidades para C-64 y C-128. Solamente en disco. Carlos Iglesias Iglesias. C/ Suárez Naranjo, 78, 6.°, D. 35004 Las Palmas de G. C.

Cambio programas (juegos, utilidades) en cinta para C-64. Por favor, enviad lista. Fco. de los Ríos Rodríguez. C/ Córdoba, 1, 2°, D. 47013. Valladolid.

Intercambio programas en disco para C-64, especialmente utilidades, gestión y aplicaciones. Poseo una programoteca bastante interesante. Francisco. Apdo. 25063. Barcelona.

Vendo Vic 20 por 15.000 pts. con cassette, manual y juegos; o bien cambiarlo por un ZX Spectrum con cassette. Alejandro Domené Pozuelo. Tl: (91) 7159990.

Intercambio programas para C-64, tanto juegos como utilidades. Daniel Vizmanos Pérez. C/ Poeta Rosalía de Castro, 25 3B. Zaragoza 50015. Tl: 514033.

Vendo C-64. Precio convenir, también cambio programas del mismo con alguien de Madrid. Manuel Rico Recondo. Tl: (91) 4414215.

Deseo formar un grupo en Cartagena y sus alrededores para cambiar juegos, información, etc. José Manuel Palao. C./ Hidalgo de Cisneros, 5, 4.°, B. 30205 Cartagena. Murcia. Tl: 519596.

1610
2030

CINCO MINUTOS ANTES DE COMPRAR UN JUEGO A 875 Ptas.

ECHALE UN VISTAZO A ESTOS JUEGOS DE 875 Ptas.



875 Ptas.
VERSION CASSETTE

COBRA SOFTWARE

SÍGUENOS EL JUEGO.

...RO SOFTWARE DIVISION Paseo de la Castellana, 141. 28046 Madrid
...459 30 04. Tel. Barna. 209 33 65. Telex: 22690 ZAFIR E